VIRTUAL INSTRUMENTATION FOR TIME SERIES ANALYSIS

by

MANOJ KUMAR

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

May, 1997

VIRTUAL INSTRUMENTATION
FOR
TIME SERIES ANALYSIS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by
MANOJ KUMAR

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
MAY, 1997

CERTIFICATE

It is certified that the work contained in this thesis entitled **"Virtual Instrumentation for Time Series Analysis"**, by Manoj Kumar, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

May 1997

Nalinaksh S Vyas
Associate Professor
Department of Mechanical Engineering
Indian Institute of Technology
Kanpur 208016

ABSTRACT

The present study aims to explore the possibility of employing the fast emerging Virtual Instrumentation (VI) technology for condition monitoring applications in rotating machinery. Time Series Analysis is known to be an effective mathematical tool, for analysis and forecasting of the behavior of dynamic systems. The objective of the present study is to translate known mathematical tools for time series analysis into Virtual Instrument by employing Graphical Programming Software, LAbVIEW The time series analysis procedure and the graphical programming features have been reviewed, briefly. Virtual Instruments have been developed to analyze standard sets of time-series data. The VIs developed, have been described in detail along with their various substructures The VIs presently incorporate Model Identification. Model Estimation and Forecasting. Features like Diagnostic checking and Noise considerations could not be incorporated into the package during the present study. These can be built as the next stage of development. to make the package complete. for on-line usage on rotating machinery

ACKNOWLEDGMENTS

With deep gratitude I express my sincere thanks to Dr N S. Vyas, my thesis supervisor, for the valuable guidance given to me. I extend my heartfelt gratitude to him for providing me an environment of complete freedom. The thesis would not have been possible had it not been for his constant encouragement.

I am highly grateful to Ms. Lalitha who devoted her valuable time in typing several drafts of the manuscript and other computer related work leading to the submission of the thesis within deadline. I gratefully acknowledge the help rendered by Mr. A.A. Khan for making diskettes and other stationaries readily available to me. Thanks are also due to Mr. Mohan Kishore for his suggestion in graphical programming whenever needed, due to Rajendra Magdum for his last minute help in setting figures, and due to Mr. M.M. Singh for making his services available whenever required.

I am thankful to one and all who were in one way or the other associated with me, towards the completion of my work. Finally, my sincere thanks to my wife for her patient understanding and support during the time I was busy with this effort.

Manoj Kumar

Indian Institute of Technology, Kanpur
May, 1997

CONTENTS

NOMENCLATURE

z_t	Time series data at time t
N	Number of time series data $(z_1, z_2, \ldots, z_N)$
$\tilde{z}_t$	Deviation of time sries data
d	Degree of differencing
w_t	Differenced time series data at time t
n	Number of data in differenced series $(w_1, w_2, \ldots\ldots w_n)$
μ	Mean of the series
K	Maximum lag
ρ	Theoretical autocorrelation coefficient
r	Estimated autocorrelation coefficient
γ	Theoretical autocovarince
c	Estimated autocorrelation
p	Order of autoregressive process
q	Order of moving average process
ϕ_{kk}	Partial autocorrelation (the last coefficient of an autoregressive process of order k)
θ_0	Constant term in the model (e.g. $\phi(B)w_t = \theta_0 + \theta(B)a_t$)
B	Backward shift operator $(Bz_t = z_{t-1})$
∇	Difference operator $(\nabla z_t = z_t - z_{t-1})$
ϕ	Autoregressive parameter
$\underline{\phi}$	Set of autoregressive parameters $(\phi_1, \phi_2, \ldots, \phi_p)$
θ	Moving average parameter
$\underline{\theta}$	Set of moving average parameters $(\theta_1, \theta_2, \ldots\ldots\ldots, \theta_q)$
a_t	Random shock (white noise)
σ_a^2	White noise variance
φ	Coefficient in difference equation used in forecasting
$\hat{z}_t(l)$	Forecast at time origin t for lead l
$e_t(l)$	Forecast error for lead time l
$V(l)$	Variance of forecast error
$\underline{\beta}$	Set of parameters $(\mu, \underline{\phi}, \underline{\theta})$, $(\mu = \overline{w})$
$\underline{\beta}_0$	Initial assumed set of parameters $(\mu, \underline{\phi}, \underline{\theta})$, $(\mu = \overline{w})$

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The idea of using a mathematical method to describe the behaviour of a physical phenomenon is well established. It is sometimes possible to derive a model based on physical laws, which enables us to calculate the value of some time dependent quantity nearly exactly, at any instant of time. If exact calculation were possible, such a model would be entirely deterministic However, no phenomenon is totally deterministic and in a variety of problems, there are many unknown factors for which it is not possible to write a deterministic model that allows exact calculation of future behaviour of the phenomenon. In such cases, it may be possible to derive a model that can be used to calculate the probability of a future value lying between two specified values limits. Such a model is called a probability model or a stochastic model.

The aim of the present study is to explore the potential offered by the fast emerging Virtual Instrumentation Technology for time series analysis Virtual Instrumentation offers the possibility of a more efficient and flexible instrumentation, data acquisition and analysis, which can be very useful for condition monitoring applications, particularly, vibration based health diagnosis of rotor-bearing systems.

Condition Monitoring and Health Diagnosis of rotating machinery in aero applications is an important area of technology development. Reference can be made to the texts by Childs (1993) and Rao (1992), for a discussion on the various aspects of rotating machinery modeling and health evaluation. The topics of mechanical fault monitoring and diagnosis have been comprehensively discussed by Mitchell (1981) and Collacott (1987).

The programming flexibility and the cost effectiveness of the Virtual Instrumentation technology, offers a possibility of more efficient and powerful vibration monitoring of rotor and forecasting of their dynamics. A traditional vibration analysis instrument is self contained, with signal input/output capabilities and fixed user interface features such as knobs, pushbuttons and other features. Specialized circuitry, including A/D converters, signal conditioning, microprocessors, memory and internal real-world signals, analyze them and present results to the user (e.g. oscilloscope). The user cannot change the instrument functionality Consider a typical rotordynamic application of forecasting the dynamics at time $t+\Delta t$ on the basis of a vibration signature measured at time t The procedure involves transferring the signal measured on a traditional oscilloscope or signal analyzer, onto a computer in digitized form, writing a computer program, obtaining the required output in numerical form and finally using a graphics package to display the forecasted dynamics. On the other hand, graphical programming in conjunction with appropriate data acquisition cards, offers the possibility of packaging the entire process above into a less expansive and more powerful virtual instrument simultaneously displaying the current and forecasted machine signature on-line and simultaneously on the computer screen.

The salient feature of a virtual instrument are

- user defined
- application oriented system with connectivity to networks, peripherals, and applications
- software based
- low cost, reusable
- open, flexible functionality based on familiar computer technology

Figures 1.1(a) and 1.1(b) depict the comparison between a traditional hardware instrument configuration and a Virtual Instrument configuration.

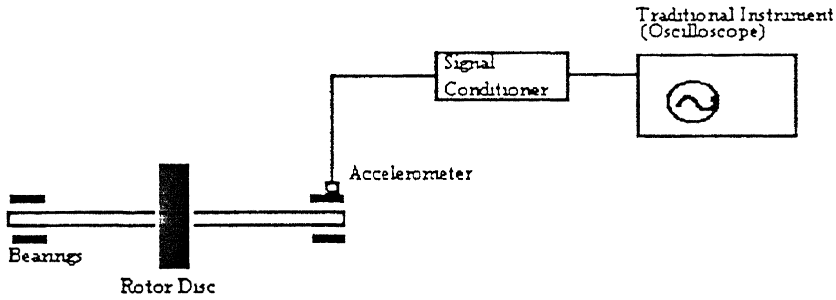


Fig. 1.1(a) Traditional Hardware Instrument configuration

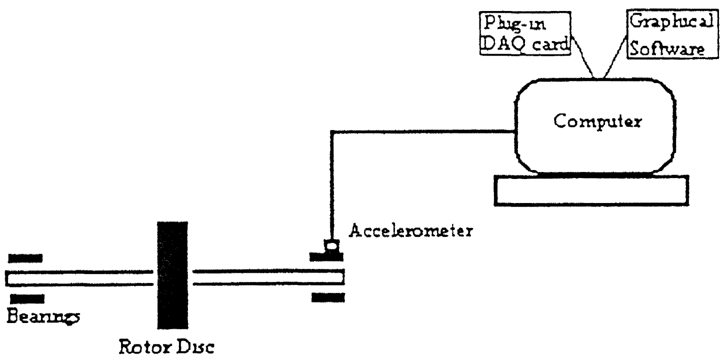


Fig. 1.1(b) Typical Virtual Instrument Configuration

The present study aims to build Virtual Instruments for typical Time Series Analysis applications, as in rotors.

Time Series Analysis involves

(i) Model Identification
(ii) Parameter Estimation
(iii) Diagnostic Checking
(iv) Forecasting

Fig. 1.2 outlines the procedure for Time Series Analysis.

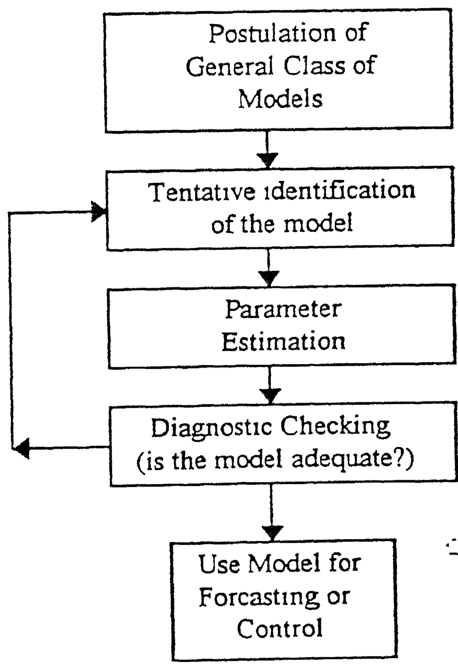


Fig. 1.2 Iterative Model Building Approach

Chapter 2 gives a brief review of Time Series Analysis procedures. The Virtual Instrumentation developed during the course of this study is described in Chapter 3. The conclusions and scope for future work is outlined in Chapter 4.

CHAPTER 2

TIME SERIES ANALYSIS: A REVIEW

The use at time t of available observations from a series in time, to forecast its value at some future time, $t+l$, finds applications in a variety of engineering, business, economic, production planning. The time period, l, over which the forecast is required is called the 'lead time'. For example, in a condition monitoring problem, the vibratory displacement reading z_t at the present hour and the readings z_{t-1}, z_{t-2}, z_{t-3} ,... ... in the previous hours might be used to forecast the vibratory displacement of the machine for lead times l = 1,2,3 ..hours ahead. The accuracy of the forecasts is expressed by calculating the *probability limits* on either side of each forecast. These limits are calculated for a convenient set of probabilities, generally between 50% to 90%.

The methods for obtaining the forecasts and estimating their probability limits are described in this chapter. A stochastic process is generally usefully defined in terms of its mean, variance and autocorrelation function. These parameters are extensively used in the present study and for the sake of completeness, their definitions are provided below.

<u>Mean:</u> The mean, μ , of a series of N observations z_1, z_2,......z_N recorded at times t_1, t_2, ...t_N is the expected value $E[z_t]$ of z_t

$\mu = E[z_t]$

and is estimated as

$$\bar{z} = \frac{1}{N}\sum_{t=1}^{N} z_t \tag{2.1}$$

<u>Autocovariance:</u> Autocovariance at lag k is defined by

$\gamma_k = \text{cov}[z_t, z_{t+k}] = E[(z_t - \mu)(z_{t+k} - \mu)]$

and is estimated as

$$c_k = \frac{1}{N}\sum_{t=1}^{N-k}(z_t - \bar{z})(z_{t+k} - \bar{z}), \qquad k = 0,1,2,\dots,K \tag{2.2}$$

<u>Autocorrelation:</u> Autocorrelation at lag k is given by

$$\rho_k = \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sqrt{E[(z_t - \mu)^2 E[(z_{t+k} - \mu)^2]}} \tag{2.3}$$

and its estimated value is given by , $r_k = c_k / c_0$

<u>Partial Autocorrelation function:</u> It is the last coefficient of an autoregressive process of order l and is estimated as

$$\begin{aligned} \phi_{ll} &= r_1 && \text{for} \quad l = 1 \\ &= (r_1 - \sum_{j=1}^{l-1}\phi_{l-1,j}r_{l-j}) / (1 - \sum_{j=1}^{l-1}\phi_{l-1,j}r_{l-j}) && \text{for} \quad l = 2,3,\dots.L \end{aligned} \tag{2.4}$$

where

$$\phi_{lj} = \phi_{l-1,j} - \phi_{ll}\phi_{l-1,l-j}\ 1 \qquad j = 1,2,\ldots\ldots,l-1 \tag{2.5}$$

2.1 REGRESSIVE PROCESSES

A linear model

$$z = \phi_1 x_1 + \phi_2 x_2 + \ldots\ldots\ldots + \phi_n x_n + a$$

relating a *dependent* variable z to a set of *independent* variables x_t plus an *error* term a, is often referred to as a *regression model*, and z is said to be regressed on x_t.

2.1.1 Autoregressive Model

In this model, the variable is regressed on previous values of itself and the current value is expressed as a finite linear aggregate of previous values of the process and a shock a_t. If the values of the process at equally spaced times $t, t-1, t-2, \ldots\ldots\ldots$ are z_t, z_{t-1}, $z_{t-2}, \ldots$ ……and $\tilde{z}_t, \tilde{z}_{t-1}, \tilde{z}_{t-2}, \ldots\ldots\ldots$ are the deviations from mean μ (e.g. $\tilde{z}_t = z_t - \mu$) then the autoregressive model is given by

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \ldots\ldots\ldots\ldots + \phi_p \tilde{z}_{t-p} + a_t \tag{2.6}$$

The above AR model is written in concise form as

$$\phi(B)\tilde{z}_t = a_t \tag{2.7}$$

where the AR operator $\phi(B)$ is

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots\ldots\ldots - \phi_p B^p \tag{2.8}$$

B is the backward operator defined as

$$B^j \tilde{z}_t = \tilde{z}_{t-j} \tag{2.9}$$

Equation (2.6) is also expressed as

$$\tilde{z}_t = \psi(B) a_t \tag{2.10}$$

where

$$\psi(B) = \phi^{-1}(B) \tag{2.11}$$

This is called an *autoregressive (AR) process of order* p, *in brief* $AR(p)$, containing $p+2$ unknown parameters μ; $\phi_1,\ldots,\phi_p$; σ_a^2 which have to be estimated from data. The additional parameter σ_a^2 is the variance of the white noise process a_t.

Autocorrelation function satisfies the difference equation

$$\phi(B)\rho_k = 0$$

i.e.

$$\rho_k = \phi_1\rho_{k-1} + \phi_2\rho_{k-2} + \ldots + \phi_p\rho_{k-p} \qquad k > 0 \tag{2.12}$$

2.1.2 Moving Average Model

In this model the current value of the process is expressed as a finite linear aggregate of current and previous shocks $a's$

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots\ldots\ldots\ldots - \theta_q a_{t-q} \tag{2.13}$$

or concisely as

$$\tilde{z}_t = \theta(B)a_t \tag{2.14}$$

where $\theta(B)$ is the MA operator given by

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots\ldots - \theta_q B^q \tag{2.15}$$

The above is called a moving average process of order q, in brief MA(q), containing $q+2$ unknown parameters μ; $\theta_1,\ldots,\theta_q$; σ_a^2.

2.1.3 Mixed Autoregressive-Moving Average (ARMA) Model

To achieve greater flexibility in fitting of actual time series, it is at times advantageous to include both AR and MA terms in the model. Thus

$$\tilde{z}_t = \phi_1\tilde{z}_{t-1} + \phi_2\tilde{z}_{t-2} + \ldots\ldots\ldots + \phi_p\tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots\ldots\ldots\ldots - \theta_q a_{t-q} \tag{2.16}$$

which can be expressed as

$$\phi(B)\tilde{z}_t = \theta(B)a_t \tag{2.17}$$

This is called Autoregreesive Moving average (ARMA) process of order (p,q). The model contains $p+q+2$ unknown parameters μ; $\phi_1,\ldots,\phi_p$, $\theta_1,\ldots,\theta_q$; σ_a^2 that are estimated from data.

2.1.4 General Autoregressive Integrated Moving Average (ARIMA) Model

This model is generally employed to analyze processes which are non-stationary in time and is written as

$$\phi(B)\nabla^d \tilde{z}_t = \theta(B)a_t \qquad (2\ 18)$$

where

$$\nabla^d \tilde{z}_t = (1-B)^d \tilde{z}_t \qquad (2.19)$$

The above is called an ARIMA(p,d,q) process which is equivalent to stationary ARMA(p,q) model

$$\phi(B)\tilde{w}_t = \theta(B)a_t \qquad (2.20)$$

where

$$\tilde{w}_t = \nabla^d \tilde{z}_t \qquad (2.21)$$

i.e. $\tilde{w}_t$ series is the d th difference of the original time series deviations $\tilde{z}_t$.

2.1.5 Time Series Analysis Procedure

The various steps in time series analysis can be listed as

i) Postulation of general class of models
ii) Tentative identification of the model and initial parameter estimation
iii) Final parameter estimation
iv) Use of model so estimated for forecasting.

A summary of the properties of autoregressive, moving average, mixed ARMA processes is given in Table 2.1.

Table 2.1 Summary of the properties of AR, MA, and Mixed ARMA processes

	Autoregressive	Moving Average	Mixed ARMA
Model in terms of previous Ξ's	$\phi(B)\Xi_t = a_t$	$\theta^{-1}(B)\Xi_t = a_t$	$\theta^{-1}(B)\phi(B)\Xi_t = a_t$
Model in terms of previous a's	$\Xi = \phi^{-1}(B)a_t$	$\Xi = \theta(B)a_t$	$\Xi_t = \phi^{-1}(B)\theta(B)a_t$
π **weights**	finite series	infinite series	infinite series
ψ **weights**	infinite series	finite series	infinite series
Stationarity condition	roots of $\phi(B)=0$ lie outside the unit circle	always stationary	roots of $\phi(B)=0$ lie outside the unit circle
Invertibility condition	always invertible	roots of $\theta(B)=0$ lie outside the unit circle	roots of $\theta(B)=0$ lie outside the unit circle
Autocorrelation function	infinite (damped exponentials and / or damped sine waves) tails off	finite cuts off at lag q	infinite (damped exponentials and / or damped sine waves) after first $q-p$ lags tails off
Partial auto-correlation function	finite cuts off	infinite (dominated by damped exponentials and /or sine waves) tails off	infinite (dominated by damped exponentials and /or sine waves after first $q-p$ lags) tails off

2.2 MODEL IDENTIFICATION

Identification methods are rough procedures applied to a set of data to indicate the kind of representational model which can be further investigated. The aim of the identification is to obtain some idea of the values of p,d and q needed to identify the general linear ARIMA (p,d,q) model to obtain initial guesses for parameters.

A general ARIMA(p,d,q) model is expressed in equation (2.17) as

$$\phi(B)\nabla^d \Xi_t = \theta(B)a_t$$

The above equation can be written as

$$\phi(B)\nabla^d z_t = \theta_0 + \theta(B)a_t$$

with

$$\theta_0 = \mu(1-\sum_{i=1}^{p}\phi_i) \qquad (2.22)$$

The following approach is used for identification

- difference z_t as many times as is needed to produce stationarity, hopefully reducing the process under study to mixed ARMA process

$$\phi(B)w_t = \theta_0 + \theta(B)a_t \qquad (2.23)$$

where

$$w_t = (1-B)^d z_t = \nabla^d z_t$$

- identify the resulting ARMA process

The above steps are carried out through use of autocorrelation functions(acf) and partial autocorrelation functions(pacf). These functions are also used at the estimation stage to provide starting values for iterative procedures employed at that stage.

2.2.1 Identification of Process Stationarity

The degree of differencing d, is identified by noting that, for a stationary mixed ARMA process of order $(p,0,q)$,

$$\phi(B)z_t = \theta(B)a_t$$

the acf satisfies the difference equation

$$\phi(B)\rho_k = 0 \quad \text{for } k > q$$

Also, if

$$\phi(B) = \prod_{i=1}^{p} (1-G_i B)$$

then solution of this difference equation for the ρ_k is of the form

$$\rho_k = A_1 G_1^k + A_2 G_2^k + \ldots\ldots + A_p G_p^k \qquad \text{for } k > q-p \qquad (2.24)$$

If the process is stationary, the requirement that the zeroes of $\phi(B)$ lie outside the unit circle (ref: Table 2.1), implies that the roots $G_1, G_2, \ldots, G_p$ lie inside the unit circle. Equation (2.24) shows that, in case of a stationary model in which none of the roots lie close to the boundary of unit circle, the acf will quickly die out for moderate and large k.

If a single real root, say G_1, approaches unity, then

$G_1 = 1-\delta$ for $(\delta << 1)$ and also positive for large k.
Therefore from (2.24), autocorrelation at lag k is

$$\rho_k \approx A_1(1-k\delta) \tag{2.25}$$

It can also be noted from Eqn.(2.25) that the acf does not die out quickly but falls off slowly and very nearly linearly. Therefore, a tendency for the acf not to die out quickly is taken as an indication that a root close to unity may exist. The estimated autocorrelation function tends to follow the behaviour of the theoretical autocorrelation function. Therefore, failure of the estimated acf to die out rapidly might logically suggest that the underlying stochastic process is to be treated as as non-stationary in z_t, but possibly as stationary in ∇z_t, or in some higher difference. It is assumed that the degree of differencing d, necessary to achieve stationarity, has been reached when the acf of $w_t = \nabla^d z_t$ dies out fairly quickly. In practice, d is either 0, 1, or 2 and is usually sufficient to inspect the first 20 or so estimated autocorrelations of the original series and of its first and second difference.

2.2.2 Stationary ARMA Process Order Identification

For an autoregressive process of order p, AR(p), whereas the autocorrelation function of an AR process tails off, its partial autocorrelation function has a cut-off after lag p. Conversely, the acf of a moving average process of order q has a cut-off after lag q, while its pacf tails off. If both the acf and pacf tail off, a mixed process is suggested. Furthermore, the acf for a mixed process, containing a pth order autoregressive component and q order moving average component, is a mixture of exponential and damped sine waves after the first $q-p$ lags. Conversely, the pacf for a mixed process is dominated by a mixture of exponentials and damped sine waves after first $p-q$ lags.
A summary of the behaviour of autocorrelation and partial autocorrelation functions is given in table 2.2.

Table 2.2 Summary Of Behaviours Of ACF And PACF For Various ARMA Models

(a)

Order	**(1,d,0)**	**(0,d,1)**
Behaviour of ρ_k	decays exponentially	only ρ_1 nonzero
Behaviour of ϕ_{kk}	only ϕ_{11} nonzero	exponential dominates decay
prelim estimate from	$\phi_1 = \rho_1$	$\rho_1 = -\theta_1/(1+\theta_1^2)$
Admissible region	$-1 < \phi_1 < 1$	$-1 < \theta_1 < 1$

(b)

Order	(2,d,0)	(0,d,2)
Behaviour of ρ_k	mixture of exponentials or damped sine wave	only ρ_1 nonzero
Behaviour of ϕ_{kk}	only ϕ_{11} and ϕ_{22} nonzero	dominated by mixture of exponentials or damped sine wave
prelim estimate from	$\phi_1 = \frac{\rho_1(1-\rho_2)}{1-\rho_1^2}$ $\phi_2 = \frac{\rho_2-\rho_1^2}{1-\rho_1^2}$	$\rho_1 = \frac{-\theta_1(1-\theta_2)}{1+\theta_1^2+\theta_2^2}$ $\rho_2 = \frac{-\theta_2}{1+\theta_1^2+\theta_2^2}$
Admissible region	$-1<\phi_2<1$ $\phi_2+\phi_1<1$ $\phi_2-\phi_1<1$	$-1<\theta_2<1$ $\theta_2+\theta_1<1$ $\theta_2-\theta_1<1$

(c)

Order	(1,d,1)
Behaviour of ρ_k	decays exponentially from first lag
Behaviour of ϕ_{kk}	dominated by exponential decay from first lag
prelim estimate from	$\rho_1 = \frac{(1-\theta_1\phi_1)(\phi_1-\theta_1)}{1+\theta_1^2-2\phi_1\theta_1}$ $\qquad \rho_2 = \rho_1\phi_1$
Admissible region	$-1<\phi_1<1$ $\qquad -1<\theta_1<1$

2.2.3 Initial Parameter Estimation Of The ARMA Process

Initial parameter estimation of an ARMA (p,q) process is based on the first $p+q+1$ autocovariances c_j $(j=0,1,\ldots,p+q)$ of $w_t = \nabla^d z_t$. It is carried out in the following three stages:

- AR parameters $\phi_1, \phi_2, \ldots, \phi_p$ are estimated from the autocovariances $c_{q-p+1}, \ldots, c_{q+1}, c_{q+2}, \ldots, c_{q+p}$, by solving the p linear equations

$$A\underline{\phi} = X \qquad \text{where}$$

$$A_{ij} = c_{|q+i-j|}\,; \qquad X_{ij} = c_{q+i} \qquad i,j = 1,2,\ldots,p \tag{2.26}$$

- Using the estimates $\phi's$ obtained in the previous step, the first $q+1$ autocovariances c'_j $(j = 0,1,\ldots,q)$ of the derived series

  $$w'_t = w_t - \phi_1 w_{t-1} - \ldots - \phi_p w_{t-p}$$

  are calculated by treating the process $w'_t = \phi(B)w_t$ as a moving average one i.e $w'_t = \theta(B)a_t$. To start with, the autocovariance c'_j of w'_t is expressed in terms of the autocovariances c_j of w_t, as

  $$c'_j = \sum_{i=0}^{p}\sum_{k=0}^{p} \phi_i \phi_k c_{|j+i-k|} \qquad p > 0 \quad (\phi_0 = -1) \tag{2.27}$$
  $$= c_j \qquad p = 0$$

- The autocovariances $c'_0, c'_1, \ldots, c'_q$ are finally used in an iterative calculation to compute initial estimates of the moving average parameters $\theta_1, \theta_2, \theta_3, \ldots, \theta_q$ and of the residual variance σ_a^2, through Newton-Raphson technique, as follows:

  Denoting the set of iterative values

  $$\tau' = (\tau_0, \tau_1, \ldots \tau_q)$$

  where $\tau_0^2 = \sigma_a^2$ and $\theta_j = -\tau_j / \tau_0 \qquad j = 1,2,\ldots,q$ (2.28)

  if τ^i is the estimate of τ, through the Newton-Raphson algorithm, obtained at ith iteration, the new values at the $(i+1)$th iteration are obtained by

  $$\tau^{i+1} = \tau^i - (T^i)^{-1} f_i \tag{2.29}$$

  where

  $$f = (f_0, f_1, \ldots, f_q),$$
  $$f_i = \sum_{i=0}^{q-j} \tau_i \tau_{i+j} - c'_j \tag{2.30}$$

  and

$$T = \begin{bmatrix} \tau_0 & \tau_1 & \cdot & \cdot & \cdot & \tau_{q-2} & \tau_{q-1} & \tau_q \\ \tau_1 & \tau_2 & \cdot & \cdot & \cdot & \tau_{q-1} & \tau_q & 0 \\ \tau_2 & \tau_3 & \cdot & \cdot & \cdot & \tau_q & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \tau_q & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \tau_0 & \tau_1 & \tau_2 & \cdot & \cdot & \cdot & \cdot & \tau_q \\ 0 & \tau_0 & \tau_1 & \cdot & \cdot & \cdot & \cdot & \tau_{q-2} \\ 0 & 0 & \tau_0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \tau_0 \end{bmatrix} \tag{2.31}$$

Iterations are carried out with starting values $\tau_0 = \sqrt{c'_0}, \tau_1 = \tau_2 = ... = \tau_q = 0$ till convergence (i.e. $|f_j|$ becomes negligible. $j = 0, 1, ..., q$).

The estimate for the overall constant, θ_0, from Eqn.(2.22), is

$$\begin{aligned} \theta_0 &= \overline{w}\left(1 - \sum_{i=1}^{p} \phi_i\right) \quad \text{for } p > 0 \\ &= \overline{w} \quad \text{for } p = 0 \end{aligned} \tag{2.32}$$

Also, the estimate of white noise variance is:

$$\begin{aligned} \sigma_a^2 &= \tau_0^2 \quad \text{for } q > 0 \\ &= c_0 - \sum_{i=1}^{p} \phi_i c_i \quad \text{for } q = 0 \end{aligned} \tag{2.33}$$

2.3 MODEL ESTIMATION

The estimation theory is based on the concept of likelihood function. If a sample of N observations is $\underline{z}$, with which an N-dimensional random variable can be associated and ξ is a set of $p+q+1$ parameters, $(\underline{\phi}, \underline{\theta}, \mu)$, the likelihood function $L(\xi|\underline{z})$ gives the conditional probability of ξ for the observed set $\underline{z}$.

2.3.1 Conditional Likelihood For An ARIMA Process

For an ARIMA model (p,d,q), the $N = n+d$ original observations $\underline{z}$, form a time series denoted by $z_{-d+1}, ..., z_0, z_1, z_2, ..., z_n$. From these observations, a series w, of $n = N - d$ differences $w_1, w_2, ..., w_n$, is generated where $w_t = \nabla^d z_t$.

The general problem of fitting parameters $\underline{\phi}$ and $\underline{\theta}$ of an ARIMA model is, then, equivalent to fitting to the $w's$, the stationary invertible ARMA(p,q) model which is written as

$$a_t = \widetilde{w}_t - \phi_1 \widetilde{w}_{t-1} - \phi_2 \widetilde{w}_{t-2} - ... - \phi_p \widetilde{w}_{t-p} + \theta_1 a_{t-1} + \theta_2 a_{t-2} + ... + \theta_q a_{t-q} \tag{2.34}$$

where

$\widetilde{w}_t = w_t - \mu$; $E[w_t] = \mu$

For $d > 0$, $\mu = 0$ is often a good approximation, otherwise $\mu = \sum_{t=1}^{n} w_t / n$.

The $w's$ cannot be substituted immediately in (2.34) because of the difficulty of starting up the difference equation (2.34). However, if p values $\underline{w}_*$ of $w's$ and q values $\underline{a}_*$ of $a's$, prior to the commencement of the w series are given, then the values of $a_1, a_2, \ldots, a_n$ conditional on this choice can be calculated in turn from equation (2.34).

For any given choice of parameters $\underline{\phi}, \underline{\theta}$ and of the starting values $\underline{w}_*$, $\underline{a}_*$ a set of values $a_t(\underline{\phi}, \underline{\theta} | \underline{w}_*, \underline{a}_*, \underline{w}_*)$ can be calculated successively. If the $a's$ are normally distributed, the joint probability density function is

$$p(a_1, a_2, \ldots, a_n) \quad \propto \quad \sigma_a^2 \exp\{-(\sum_{t=1}^{n} a_t^2 / 2\sigma_a^2)\}$$

Taking log on both sides of the above proportionality, the log likelihood associated with the parameter values $(\underline{\phi}, \underline{\theta}, \sigma_a)$, for a particular set of data $\underline{w}$, conditional on the choice of $(\underline{w}_*, \underline{a}_*)$ is obtained as

$$l_*(\underline{\phi}, \underline{\theta}, \sigma_a) = -n \ln \sigma_a - \frac{S_*(\underline{\phi}, \underline{\theta})}{2\sigma_a^2} \tag{2.35}$$

where $$S_*(\underline{\phi}, \underline{\theta}) = \sum_{t=1}^{n} a_t^2(\underline{\phi}, \underline{\theta} | \underline{w}_*, \underline{a}_*, \underline{w}) \tag{2.36}$$

It can be noted from Eqn.(2.35) that lower the $S_*(\underline{\phi}, \underline{\theta})$, higher will be the log likelihood function. Also data are involved only through S_*, the sum of squares. In order to get the maximum likelihood function, the sum of the square function $\sum_{t=1}^{n} a_t^2$ is minimized. The minimized S_*, called least square estimate, corresponds to the desired set of parameters.

2.3.2 Nonlinear Estimation

For most situations, the maximum likelihood estimates are closely approximated by the least square estimates, which make

$$S_*(\underline{\phi}, \underline{\theta}) = \sum_{t=-\infty}^{n} [a_t | \underline{\phi}, \underline{\theta}, \underline{w}]^2 = \sum_{t=-\infty}^{n} [a_t]^2$$

a minimum.

In practice, the infinite sum can be replaced by a finite sum $\sum_{t=-Q}^{n} [a_t]^2$.

$$a_t = (w_t - \mu) - \sum_{i=1}^{p} \phi_i (w_{t-i} - \mu) + \sum_{j=1}^{q} \theta_j a_{t-j} \tag{2.41}$$

- For the given values of the parameters $(\mu, \underline{\phi}, \underline{\theta})$ the residual sum of squares is calculated from $\sum_{t=-Q}^{n} [a_t]^2$.
- The least squares estimates are calculated next. The values of the parameters which minimize the residual sum of squares are obtained by a constrained optimization method (Marquardt method). Denoting all the initial assumed set of parameters $(\mu, \underline{\phi}, \underline{\theta})$ by

  $\underline{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ where $k = p + q + 1$ (2.42)

  the derivative $x_{i,t}$ is calculated from Eqn.(2.39)

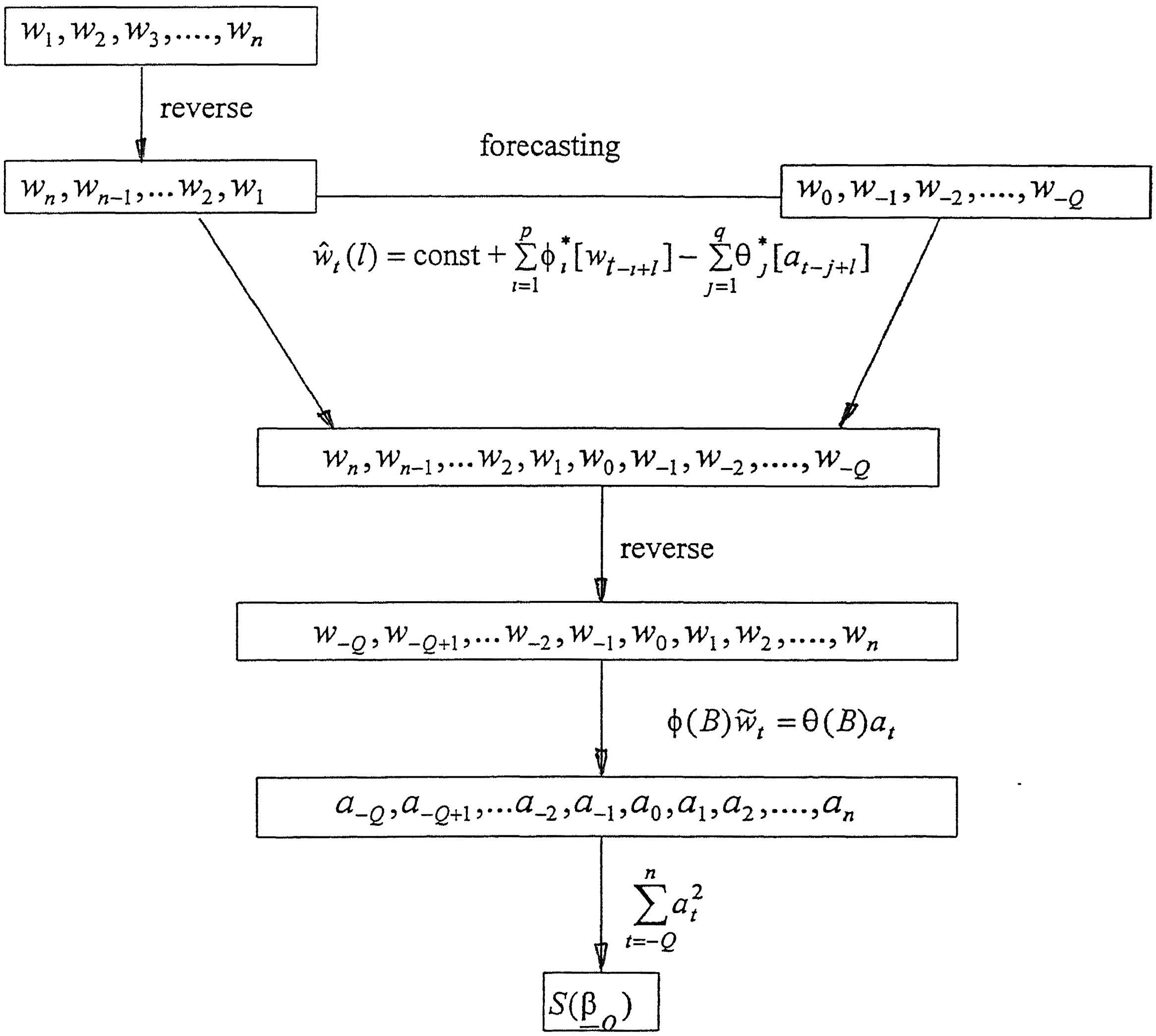


Fig. 2.1 Backforecasting $\tilde{w}$ series

Minimization is carried out in three iterative stages:

Stage(1)

(1) With a_t, $x_{i,t}$, the $k \times k$ matrix $A = \{A_{ij}\}$ is calculated as

$$A_{ij} = \sum_{t=-Q}^{n} x_{i,t} x_{j,t} \tag{2 43}$$

(2) The vector g with elements $g_1, g_2, \ldots\ldots, g_k$ is calculated as

$$g_i = \sum_{t=-Q}^{n} x_{i,t} a_t \tag{2.44}$$

(3) Scaling quantities (for future use) are computed as

$$D_i = \sqrt{A_{ii}} \tag{2.45}$$

Stage(2)

(1) The modified (scaled and constrained) linearized equations

$$A^* \, h^* = g^* \tag{2.46}$$

are formed according to

$$\begin{aligned} A_{ij}^* &= A_{ij} / D_i D_j \qquad i \neq j \\ A_{ii}^* &= 1 + \pi \\ g_i^* &= g_i / D_i \end{aligned} \tag{2.47}$$

(2) The equations are solved for h^* which is scaled back to give the parameter corrections

$$h_j = h_j^* / D_j \tag{2.48}$$

(3) The new parameter values are calculated using

$$\underline{\beta} = \underline{\beta}_0 + \underline{h} \tag{2.49}$$

which is used in computing the next sum of squares $S(\underline{\beta})$.

Stage(3)

(1) If $S(\underline{\beta}) < S(\underline{\beta}_0)$ and the parameter corrections $\underline{h}$ are all negligible then convergence is assumed. Otherwise, next iteration is started from Stage(1), with the new parameter values $\underline{\beta}$ and the reduced value of π by a factor 10.

(2) If $S(\underline{\beta}) > S(\underline{\beta}_0)$, the parameter π is increased by a factor 10 and computation is started at Stage(2).

2.4 FORECASTING

A general ARIMA process can be forecast in three different explicit forms:

1. in terms of previous values of the $z's$ and current and previous values of the a's by direct use of the difference equation.
2. in terms of current and previous shocks a_{t-j} only.
3. in terms of a weighted sum of previous values z_{t-j} of the process and the current shock a_t.

Forecasting is done in the present study using the difference equation form (1 above), of the model, which is

$$z_t = \varphi_1 z_{t-1} + \ldots\ldots + \varphi_{p+d} z_{t-p-d} - \theta_1 a_{t-1} - \ldots\ldots - \theta_q a_{t-q} + a_t \qquad (2.50)$$

The value z_{t+l}, $l \geq 1$ is forecasted at the time origin t. This forecast is said to be made at origin t for lead time l, as follows

$$z_{t+l} = \varphi_1 z_{t+l-1} + \ldots\ldots + \varphi_{p+d} z_{t+l-p-d} - \theta_1 a_{t+l-1} - \ldots\ldots - \theta_q a_{t+l-q} + a_{t+l} \qquad (2.51)$$

2.4.1 Minimum Mean Square Error Forecasts

The minimum mean square error forecast at origin t, for lead time l is the conditional expectation of z_{t+l} at time t (Box and Jenkins, 1976).

Mathematically, by regarding $\hat{z}_t(l)$ as the forecast function of l for a fixed origin t,

$$\begin{aligned} z_t(l) &= [z_{t+l}] \\ &= E[z_{t+l} | z_t, z_{t-1}, \ldots.] \\ &= \varphi_1 [z_{t+l-1}] + \ldots + \varphi_{p+d} [z_{t+l-p-d}] - \theta_1 [a_{t+l-1}] - \ldots - \theta_q [a_{t+l-q}] + [a_{t+l}] \\ &= \sum_{i=1}^{p+d} \varphi_i [z_{t-i+l}] - \sum_{j=1}^{q} \theta_j [a_{t-j+l}] \end{aligned} \qquad (2.52)$$

where

$$\begin{aligned} [z_{t-j}] &= E[z_{t-j}] = z_{t-j} && j = 0,\ 1,\ 2\ldots.. \\ [z_{t+j}] &= E[z_{t+j}] = \hat{z}_t(j) && j = 1,\ 2\ldots\ldots. \\ [a_{t-j}] &= E[a_{t-j}] = a_{t-j} = z_{t-j} - \hat{z}_{t-j-1}(1) && j = 0,\ 1,\ 2,\ldots \\ [a_{t+j}] &= E[a_{t+j}] = 0 && j = 1,\ 2,\ldots\ldots \end{aligned} \qquad (2.53)$$

For moving average terms in the difference equation, the forecasting process is started off initially by setting unknown $a's$ equal to their unconditional expected values of zero. In general, if the moving average operator $\theta(B)$ is of degree q, the forecast equations for $\hat{z}_t(1)$, $\hat{z}_t(2)$,.......$\hat{z}_t(q)$ depend directly on the $a's$ but forecast at longer lead times do not.

2.4.2 Forecast Errors

The forecast error for lead time l and its variance are

$$e_t(l) = a_{t+l} + \psi_1 a_{t+l-1} + \ldots + \psi_{l-1} a_{t+1}$$
$$V(l) = \text{var}[e_t(l)] = (1 + \psi_1^2 + \psi_2^2 + \ldots\ldots\ldots + \psi_{l-1}^2)\sigma_a^2 \tag{2.54}$$

The weights, $\psi_1, \psi_2, \ldots, \psi_{l-1}$ in equations (2.54) are calculated by equating coefficients of powers of B in

$$(1 - \varphi_1 B - \ldots - \varphi_{p+d} B^{p+d})(1 + \psi_1 B + \psi_2 B^2 + \ldots) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q$$

to get

$$\psi_j = 1 \qquad j = 0 \tag{2.55}$$
$$= \sum_{l=1}^{j} \varphi_i \psi_{j-i} - \theta_j \qquad j \geq 1$$

2.4.3 Probability Limits Of The Forecasts

Since the square of the error between the forecasted value for lead l and the actual value is

$e_t^2(l) = \{z_{t+l} - \hat{z}_t(l)\}^2$.

If the $a's$ are normal, the conditional probability distribution $p(z_{t+l}|z_t, z_{t-1}, \ldots\ldots\ldots)$ of a future value z_{t+l} of the process will be Normal with mean $\hat{z}_t(l)$ and standard deviation

$$V(l) = \left\{1 + \sum_{j=1}^{l-1} \psi_j^2\right\} \sigma_a^2$$

The upper and lower probability limits are, therefore, obtained by (Box & Jenkins, 1976)

$$z_{t+l}(\pm) = \hat{z}_t(l) \pm u\sqrt{V(l)} \tag{2.56}$$

where

$u = 0.68$ for 50 % probability
$= 1.65$ for 90 % probability
$= 1.96$ for 95 % probability
$= 2.58$ for 99 % probability

The forecast is updated with the arrival of every new data z_{t+1}, by updating the origin to $t+1$ and by calculating the new forecast error.

The Virtual Instruments developed on the basis of the Time Series Analysis procedure described here are discussed in Chapter 3.

CHAPTER 3

VIRTUAL INSTRUMENTATION

The original plan, in the present study, was to develop Virtual Instruments for Model Identification, Parameter Estimation, Diagnostic Checking and Forecasting of the dynamics of an available laboratory rotor rig. However, due to time constraints the attempt had to be restricted to off-line development of Virtual Instrumentation for general Time Series Analysis. The VIs have been tested and validated, on the basis of available literature and are in a state, where they can be implemented on the rotor-rig vibration signals, with little effort. The rotor-rig, plug-in data acquisition card and the software are briefly described in the following.

3.1 THE ROTOR RIG

The available laboratory rotor rig comprises of a shaft supported in bearings (with provision for rolling element or fluid film bearings), with discs and flexible couplings.It can be employed to simulate a variety of real life rotor dynamic problems (e.g. unbalance, oil whirl, shaft bow, cracks, bearing wear and tear etc.), refer Figure 3.1. Vibration signals can be picked up through accelerometers (or proximity pick ups). Signal conditioning is done through Charge Amplifiers. The conditioned and amplified signals from Charge Amplifiers can be taken into the available PC486, through AD/DA conversion cards.

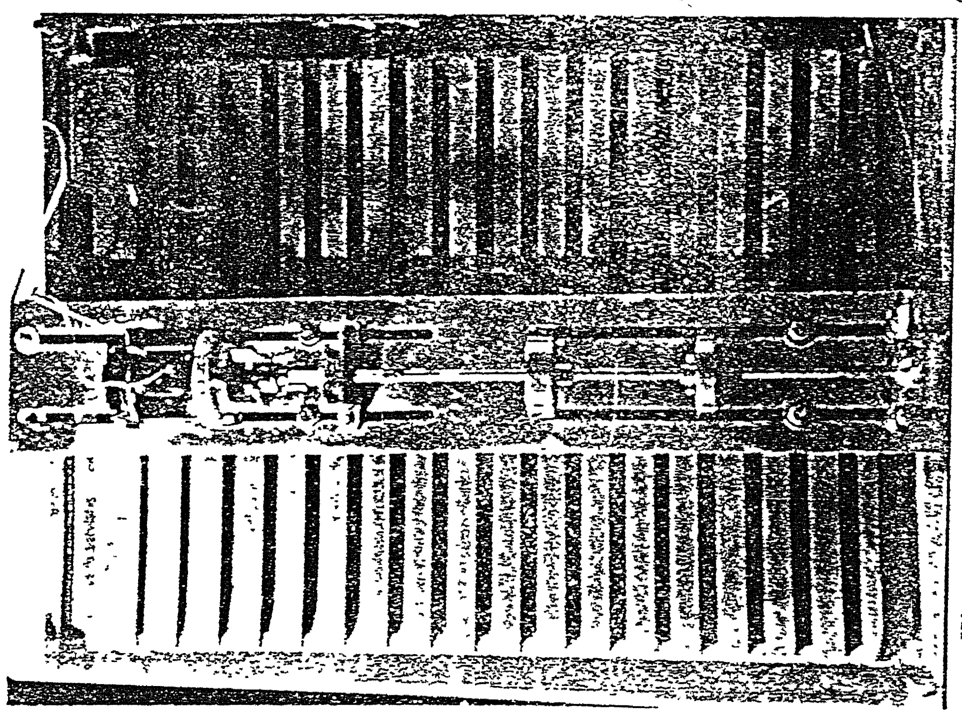


Fig. 3.1 Rotor Rig

3.2 PLUG IN DATA ACQUISITION

A Plug-in DAQ board, AT-MIO-16E-10, from National Instruments, Texas, has been acquired. This board has various combinations of analog, digital, and timing inputs and outputs The board can acquire data in three modes: 1) continuous acquisition of a single channel, 2) multichannel acquisition with continuous scanning, or 3) multichannel acquisition with interval scanning. There are sixteen input channels and eight output channels

3.3 SOFTWARE

A graphical software, LabVIEW has been acquired from National Instruments, Texas, USA. LabVIEW is a program development application, much like various commercial C or BASIC development systems While other programming systems use text based languages to create the code, LabVIEW uses a graphical programming language, G, to create programs in block diagrams. It uses icons and graphical symbols to describe programming action. LabVIEW contains functions and subroutines for various programming tasks, like in other languages. LabVIEW programs are called Virtual Instruments (VIs) because their appearance and operation imitate actual instruments. They are, however, analogous to functions from conventional language programs. VIs have both an interactive user interface and source code equivalent and accept parameters from higher level VIs The following are the descriptions of the salient VI features.

- The interactive user interface of a VI is called Front Panel, because it simulates the panel of a physical instrument. The front panel can contain graphs, push buttons, knobs and other control indicators. Data can be input using a mouse or a keyboard.
- VI receives instructions from a block diagram which is a graphical program written in G The block diagram is also the source code.
- VIs are hierarchical and modular and can be used as top-level programs or as subprograms within other programs or subprograms. A VI within a VI is called a subVI.

The variety of functions available in LabVIEW are described below.

Arithmetic functions : All types of arithmetic functions like add, subtract, multiply, round to nearest, increment, decrement, scale by, power of 2, quotient and remainder round to + infinity, round to - infinity, complex conjugate, absolute value, negate, square toot, reciprocal, random number etc.
Logical functions : AND, OR, EXCLUSIVE OR, NOT, NOR, NAND etc.
Trignometric and logarithmic functions : Sine, Cosine, tangent, inverse sine etc.
Comparison function : Equal to, not equal, greater than, lesser than, max and min,greater than 0, not equal to zero etc.
Conversion functions : To byte integer, to word integer, to long integer, to unsigned byte, complex to polar, cluster to array etc.
String function : String length, concatenate, string, string subset, split string , to decimate etc.

Array Functions : *array size* used to find the length of an array, *index array* to extract an element at the given index, *replace array element* to replace the existing element with the desired element, *array subset* to extract subset of given length of rows and columns, *build array* to build an array of elements etc.

LabVIEW has two structures to repeat execution of a subdiagram - the While Loop and the For Loop

While Loop · This is a post-iterative test loop structure that repeats a section of code until a condition is met (Fig 3.2(a)). It is comparable to a Do loop or a Repeat-Until loop in conventional programming languages.

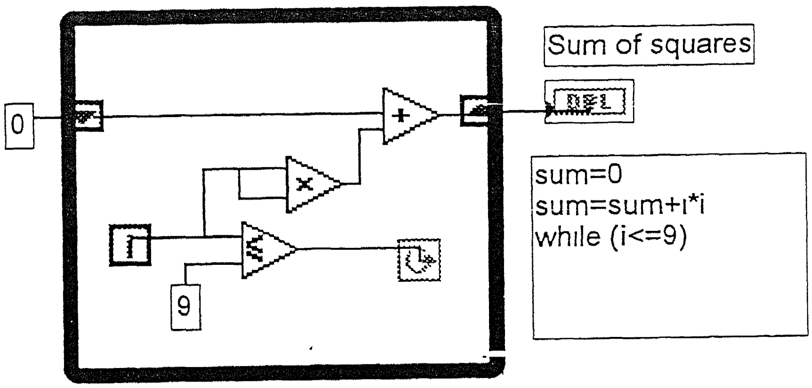


Fig. 3.2(a) Block diagram for While loop

The For Loop : This loop (Fig. 3.2(b)) executes the diagram inside its border a predetermined number of times. It has two terminals : the count terminal (an input terminal) and the iteration terminal.The count terminal specifies the number of times to execute the loop.The iteration terminal contains the number of completed iterations. The For Loop is equivalent to

For i = 0 to N - 1

Execute Diagram Inside The Loop

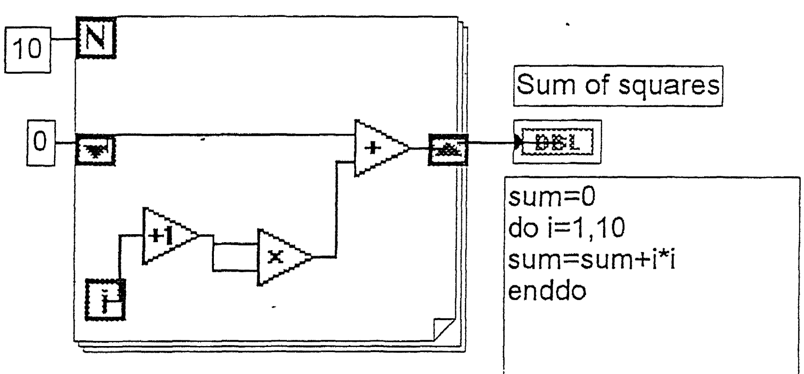


Fig. 3.2(b) Block diagram for loop

LabVIEW has two structures that control data flow. They are the Case structure(analogous to'if-then-else' statement in a conventional text based language) and the Sequence structure

Case structure . This is a conditional branching control structure (Fig. 3.2(c)), which executes one and only one condition of its subdiagrams based on its input. It is the combination of the IF, THEN, ELSE, and CASE statements in control flow languages.

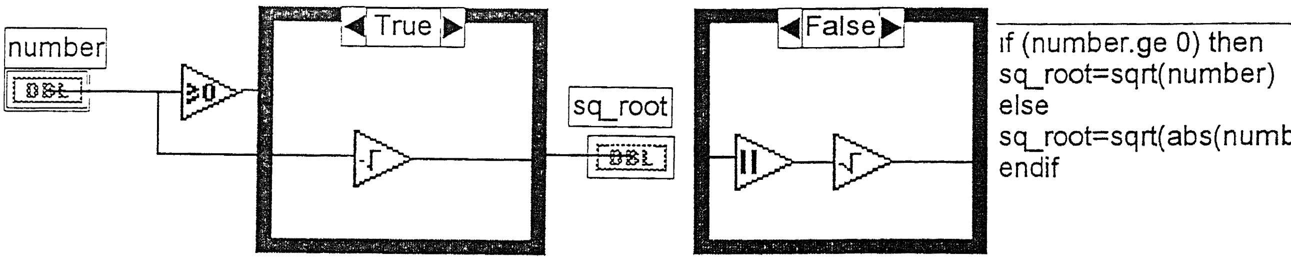


Fig. 3.2(c) Block diagram for Case structure

Sequence structure : This is a program control structure that executes its subdiagrams in a specific, numeric order. It is commonly used to force nodes that are not data-dependent to execute in a desired order. The portion of the diagram to be executed first is placed in frame 0 of the Sequence structure, the diagram to be executed second is placed in frame 1, and so on Frame is subdiagram of a Sequence structure and only one frame is visible at a time.

Some other relevant features of the software are -

Strings: A string is a collection of ASCII characters. A numeric data can be passed as character strings and then these are converted to numbers.

File I/O : These are functions for working with files. In addition to reading and writing data, the LabVIEW file I/O functions move and rename files and directories, create spreadsheet-type files of readable ASCII text, and write data in binary form for speed and compactness.

Code Interface Node (CIN): CIN is a node which links external code written in a conventional programming language to LabVIEW. A CIN appears on the diagram as a set of input and output terminals.

3.4 COMPUTER PROGRAMS

Virtual instruments have been developed for six sets of time series data available in literature (Box and Jenkins, 1976). This is necessary to validate and check the graphical programs before their on-line application to the rotor-rig. The six time series' taken for this purpose pertain to

Series A.	Chemical process concentration readings, every two hours
Series B	IBM common stock closing prices, daily, 17th May '61- 2nd November '62
Series C:	Chemical process temperature readings, every minute.
Series D:	Chemical process viscosity readings, every hour
Series E.	Wolfer sunspot numbers, yearly
Series F:	Series of 70 consecutive yields from a batch chemical process

These sets of data have been chosen for programming and validation for they represent a wide spectrum of *known* stochastic behaviour. Series A is a typical ARMA(1,0,1) process. Series B is an Integrated Moving Average, IMA(0,1,1) process. Series C is an Integrated Autoregressive, IAR(1,1,0) process. Series' D,E and F are pure AR(1,0,0), AR(2,0,0), AR(2,0,0) processes respectively.

The overall picture of the VIs developed can be seen from Figures 3.3-3.8. These figures show the front panels of the Virtual Instruments developed, for Series A-F, respectively. The panels simultaneously show the input data, the autocorrelations, the partial autocorrelations, the forecast and the forecast limits. Some of the estimated parameters can also be seen in the form of pellets. The forecast are shown for a probability of 50%.

Figures 3.9(a) and 3.9(b) typically show the graphical program (called the Block Diagram), in a condensed form, for the front panels of Fig. 3.3 (Series A). This main VI contains several subVIs. Figure 3.10 describes the position of these subVIs in the hierarchy of computational operations. Figure 3.11 gives a consolidated list of these subVIs.

The various subVIs which are used in the main VI are described below.

del.vi (Fig.3.12):	differences the original time series (input data) $\underline{z}=\{z_1,z_2,..,z_N\}$ to get $\underline{w}=\{w_1,w_2,....w_{N-d}\}$ according to the relation $w_t = \nabla^d z_t$
c & r.vi (Fig.3.13)	calculates the autocovariance and autocorrelation coefficients according to Eqns.(2.2) and (2.3).
pacf.vi (Fig.3.14)	represents the graphical program to calculate partial autocorrelation according to Eqns.(2.4)-(2 5).

autoreg_para.vi (Fig.3.15)	calculates initial autoregressive parameters according to Eqns.(2 26).
prelim parameter estimation1.vi (Fig.3.16)	Figs 3 16(a)-(f) together with Fig.3.15 represent the graphical program for preliminary estimation of autoregressive and moving average parameters at the tentative identification stage according to Eqns.(2.27)- (2.31).
phi to phi*.vi (Fig.3.17)	Figs.3.17(a)-(b) represent the program to convert ϕ to φ according to $\varphi(B) = \phi(B)(1-B)^d$
forecastA.vi (Fig 3 18)	calculates forecasts of time series at given origin t for lead l according to Eqns. (2.51)-(2 53) (used as a subVI of Fig 3.19)
back_fore.vi (Fig 3.19)	calculates backforecasts of $\overline{w}$ series before starting the recursive calculation of a_t
a(t) & sum of sq.vi (Fig 3.20)	recursively calculates a_t and its sum of squares S according to Eqns.(2.41) and (2.36) at the model estimation stage.
a(t) & residual sum of squares.vi (Fig.3.21).	This is a program to recursively calculate a_t and its sum of squares S, according to Eqns.(2.41) and (2.36) at the model estimation stage. This VI uses Fig.3.19 and Fig.3.20 as subVIs.
final estimation icon.vi (Fig.3.22)	Figs.3.22(a)-(g) represent the graphical program in LabVIEW incorporating the model estimation algorithm described in Eqns. (2.38), (2.41)- (2.49). This program uses the subVIs of Fig.3.21
forecast.vi (Fig 3 23)	Figs. 3.23(a)-(i) forecasts the final model according to the algorithm described in Eqns. (2.50)-(2.56).

3.5 RESULTS

The results obtained for the Series A, after implementation of the VIs described above are discussed here

Model Identification of series A The behaviour of autocorrelation and partial autocorrelation functions are studied, as discussed in Chapter 2, to identify a general ARIMA (p,d,q) model. The autocorrelation and partial autocorrelation coefficients obtained fiom the VIs of Figs.3 13 and 3.14 are given in Tables 3.1 - 3 2 and Figs. 3.24 - 3.25

Table 3.1 Estimated Autocorrelations for series A

Lag k	d=0	d=1	d=2
1	0.57	-0.41	-0.65
2	0 50	0.02	0.18
3	0.40	-0.07	-0.04
4	0.36	-0.01	0.04
5	0.33	-0.07	-0.04
6	0.35	-0.02	-0.04
7	0 39	0.15	0.13
8	0.32	-0.07	-0.11
9	0.30	0.04	0.04
10	0.26	0.02	0.02
11	0.19	-0.05	-0.02
12	0 16	-0.06	-0.02
13	0.20	-0.01	-0.04
14	0.24	0.16	0.18
15	0.14	-0.17	-0.19
16	0 18	0.03	0.08
17	0 20	0.01	-0.03
18	0.20	0.08	0.09
19	0 14	-0.12	-0.17
20	0.18	0.15	0.20

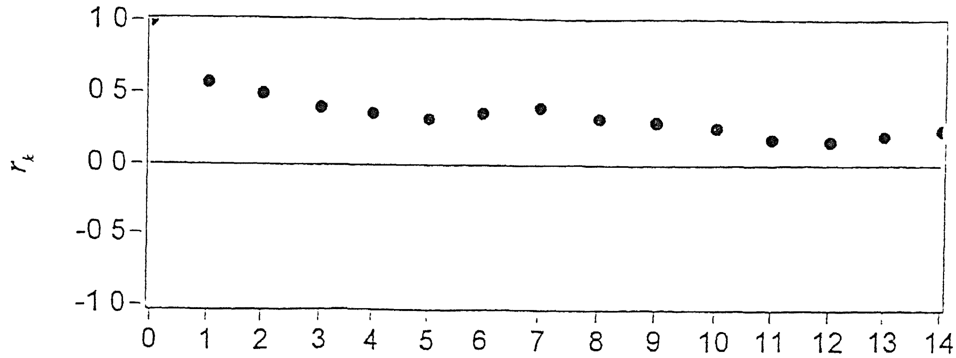


Fig. 3.24(a) Autocorrelation for $d=0$

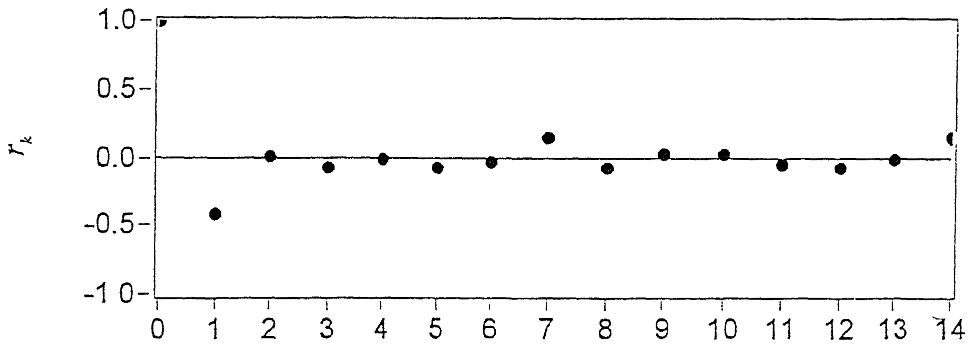


Fig. 3.24(b) Autocorrelation for $d=1$

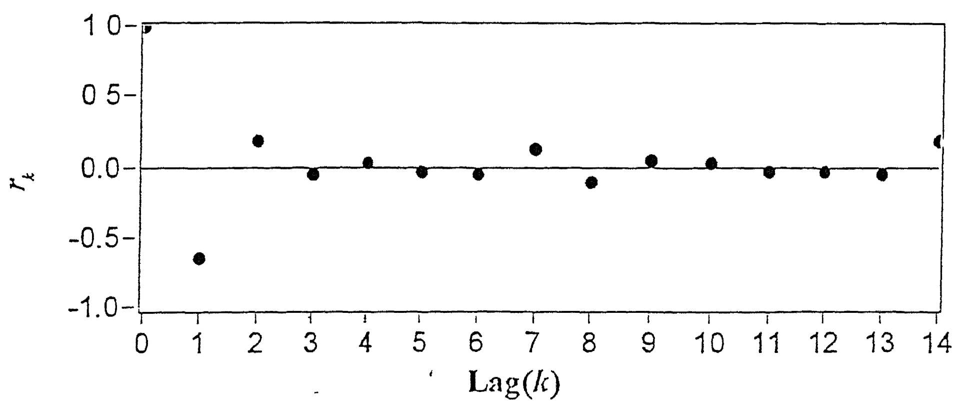


Fig. 3.24(c) Autocorrelation for $d=2$

Table 3.2 **Estimated Partialautocorrelations ϕ_{kk} For Series A**

Lag k	d=0	d=1	d=1
1	0.57	-0.41	-0.65
2	0.25	-0.18	-0.42
3	0 07	-0.17	-0.32
4	0.07	-0.14	-0 20
5	0.12	-0.19	-0.18
6	0.16	-0.21	-0 31
7	-0 03	0 00	-0.18
8	0.01	-0 05	-0 15
9	-0 02	-0.02	-0.15
10	-0.07	0.04	-0 05
11	-0 02	-0.01	0.02
12	-0 02	-0.08	0.02
13	0 06	-0.10	-0 16
14	0 09	0 10	0 05
15	-0 12	-0 09	0.06
16	0.05	-0 13	0 00
17	0 10	-0.10	-0 12
18	0 07	0.05	0 01
19	-0.07	-0.07	-0.12
20	0.05	0 09	0.07

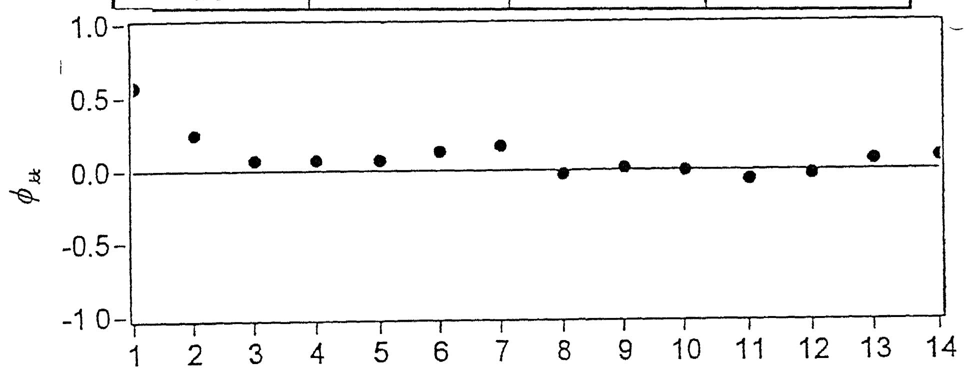


Fig. 3.25(a) **Partial autocorrelation for $d = 0$**

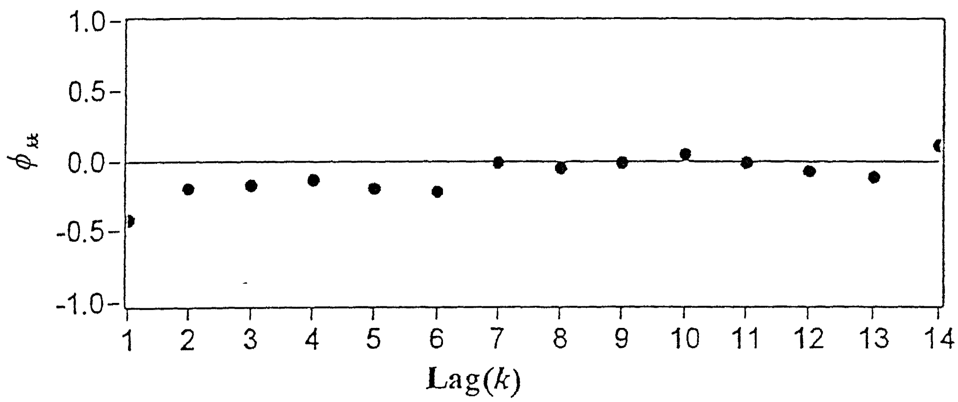


Fig. 3.25(b) **Partial autocorrelation for $d = 1$**

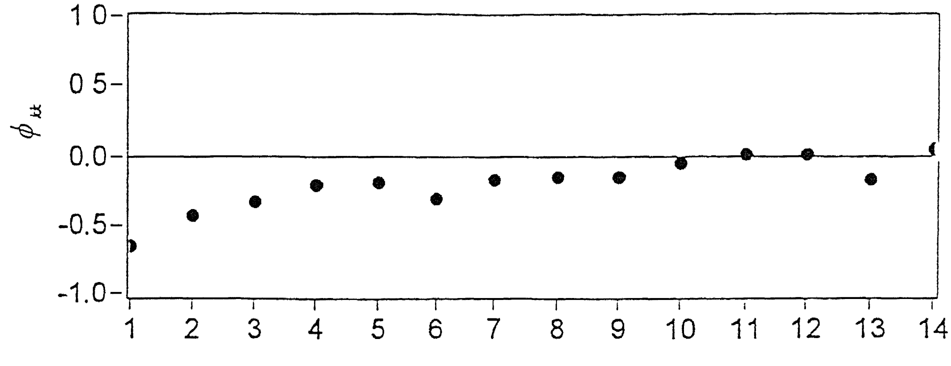


Fig. 3.25(c) Partial autocorrelation for $d = 2$

It is observed that the autocorrelation for $d = 1$ are small after the first lag, suggesting that this series might be described by an IMA(0,1,1) process. However, from the autocorrelation function for $d = 0$, it is seen that after lag 1, autocorrelation decreases fairly regularly. Therefore, an alternative possibility is that the series is a mixed ARMA of order (1,0,1). The partial autocorrelation function for $d = 0$ tends to support this possibility. Thus, for further analysis of series A, these two possible models IMA (0,1,1) and ARMA(1,0,1) are considered.

<u>Initial Parameter Estimation</u>: With tentative models IMA (0,1,1) and ARMA(1,0,1) identified, the initial estimates of θ and ϕ are calculated from Eqns.(2.25)-(2.34). The estimates are given in Table 3.3.

Table 3.3 Initial Estimates For Series A

Model	ϕ	θ	σ_a^2	θ_0
(1,0,1)	0.87	0.48	0.098	2.25
(0,1,1)	-	0.53	0.107	0 00

<u>Final Estimation</u>: With the tentative model identification and initial estimates of parameters, more efficient estimates are calculated for the series, by using the initial models and parameters as initial guesses, e.g. for IMA(0,1,1), the initial guess is $\underline{\beta}_0 = (\theta, \mu)$ where $\theta = 0.53$ and $\mu = 0$.

Nonlinear least square estimates are, now calculated using constrained optimization (Marquardt Method) by Eqns.(2.38), (2.41)-(2 49) and the graphical program of Figs 3.21(a)-(g). The final estimates are given in Table 3.4

Table 3.4 Final Estimates For Series A

Model	ϕ	θ	σ_a^2	θ_0
(1,0,1)	0.91	0.56	0 099	1.62
(0,1,1)	-	0.71	0.102	0.00

Forecasting:

Forecasts at origin 144 with lead $l = 1,2,...,53$ are calculated, for ARMA(1,0,1) possibility, according to Eqn.(2 52) The values of upper and lower limits for 50% probability are calculated from Eqns.(2.53)-(2.56). The actual data, along with the corresponding forecast and the upper and lower limits are given in Table 3.5.

Table 3.5 Forecasting At Origin 144 For 50% Probability Limit

Actual Data	Forecast	Upper Limit	Lower Limit
16.70	16.73	16.95	16.52
16.90	16.77	16.99	16.54
17.40	16.79	17.03	16.56
17.10	16.82	17.06	16.57
17.00	16.84	17.09	16.59
16.80	16.86	17.12	16.61
17.20	16.88	17.14	16.62
17.20	16.90	17.16	16.64
17.40	16.91	17.18	16.65
17.20	16.93	17.19	16.66
16.90	16.94	17.21	16.67
16.80	16.95	17.22	16.68
17.00	16.96	17.23	16.69
17.40	16.97	17.24	16.70
17.20	16.98	17.25	16.71
17.20	16.99	17.26	16.71
17.10	16.99	17.27	16.72
17.10	17.00	17.27	16.73
17.10	17.01	17.28	16.73
17.40	17.01	17.29	16.74
17.20	17.02	17.29	16.74
16.90	17.02	17.29	16.75

16.90	17.02	17.30	16.75
17.00	17.03	17.30	16.75
16.70	17.03	17.31	16.76
16.90	17 03	17.31	16.76
17.30	17 04	17.31	16.76
17.80	17.04	17.31	16.76
17.80	17 04	17.32	16.76
17.60	17.04	17.32	16.77
17.50	17 04	17.32	16.77
17.00	17.05	17.32	16.77
16.90	17.05	17.32	16.77
17.10	17.05	17.32	16.77
17.20	17.05	17.32	16.77
17.40	17.05	17.33	16.77
17.50	17.05	17.33	16.78
17.90	17.05	17.33	16.78
17.00	17.05	17.33	16.78
17.00	17 05	17.33	16.78
17.00	17.05	17.33	16.78
17.20	17 05	17.33	16.78
17.30	17 06	17.33	16.78
17.40	17.06	17.33	16.78
17.40	17.06	17.33	16.78
17.00	17.06	17.33	16.78
18.00	17 06	17.33	16.78
18.20	17.06	17.33	16.78
17.60	17.06	17.33	16.78
17.80	17.06	17.33	16.78
17.70	17 06	17.33	16.78
17.20	17.06	17.33	16.78
17.40	17.06	17.33	16.78

Similarly the remaining series' were analysed. Summary of models identified for series A-F in terms of various initial parameter estimates are given in Table 3.6.
Final estimates for all 6 series' are given in Table 3.7.

Table 3.6 Initial Estimates For Series A-F

Series	Model	$\underline{\phi}$	$\underline{\theta}$	σ_a^2	θ_0
A	(1,0,1)	$\phi_1 = 0\ 87$	$\theta_1 = 0\ 48$	0.098	2.25
B	(0,1,1)	-	$\theta_1 = -0\ 09$	52 151	-0.28
C	(1,1,0)	$\phi_1 = 0\ 81$	-	0.019	-0.01
D	(1,0,0)	$\phi_1 = 0.86$	-	0.093	1.27
E	(2,0,0)	$\phi_1 = 1\ 32$ $\phi_2 = -0\ 63$	-	289.214	14 86
F	(2,0,0)	$\phi_1 = -0\ 32$ $\phi_2 = 0\ 18$	-	114.719	58.29

Table 3.7 Final Estimates For Series A-F

Series	Model	$\underline{\phi}$	$\underline{\theta}$	σ_a^2	θ_0
A	(1,0,1)	$\phi_1 = 0\ 91$	$\theta_1 = 0\ 56$	0.099	1.62
B	(0,1,1)	-	$\theta_1 = -0.09$	52.438	-0.28
C	(1,1,0)	$\phi_1 = 0.82$	-	0.018	-0.01
D	(1,0,0)	$\phi_1 = 0.87$	-	0.091	1.17
E	(2,0,0)	$\phi_1 = 1\ 42$ $\phi_2 = -0\ 73$	-	234.867	14.72
F	(2,0,0)	$\phi_1 = -0\ 35$ $\phi_2 = 0\ 19$	-	117.751	59.05

3.6 REMARKS

The numerical results obtained above, for the six sets of time series data are compared with those given by Box and Jenkins (1976). A good agreement has been found for the forecasts and intermediate parameter. This is an indication of the correctness of the VIs developed

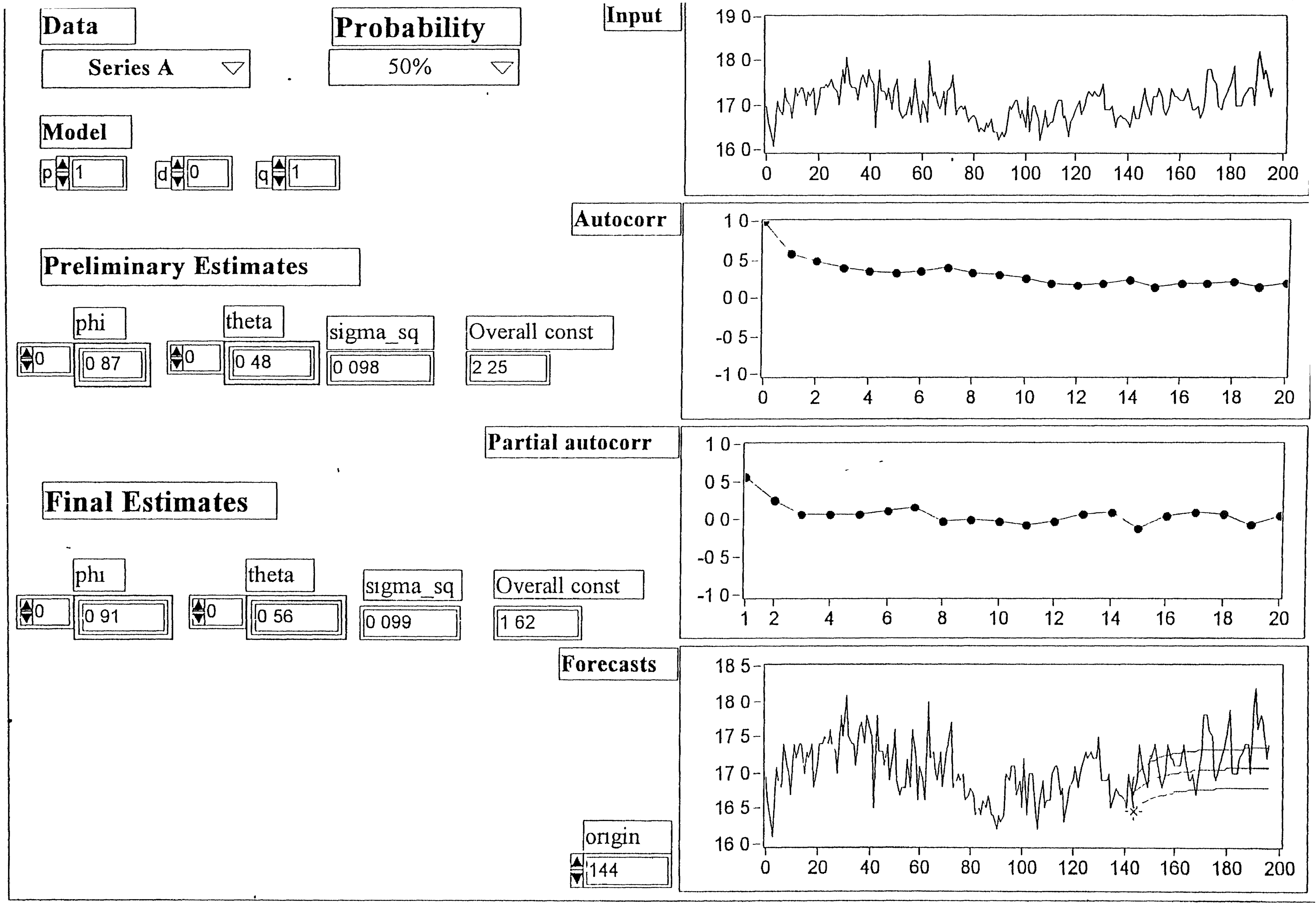


Fig. 3.3 Front Panel for series A

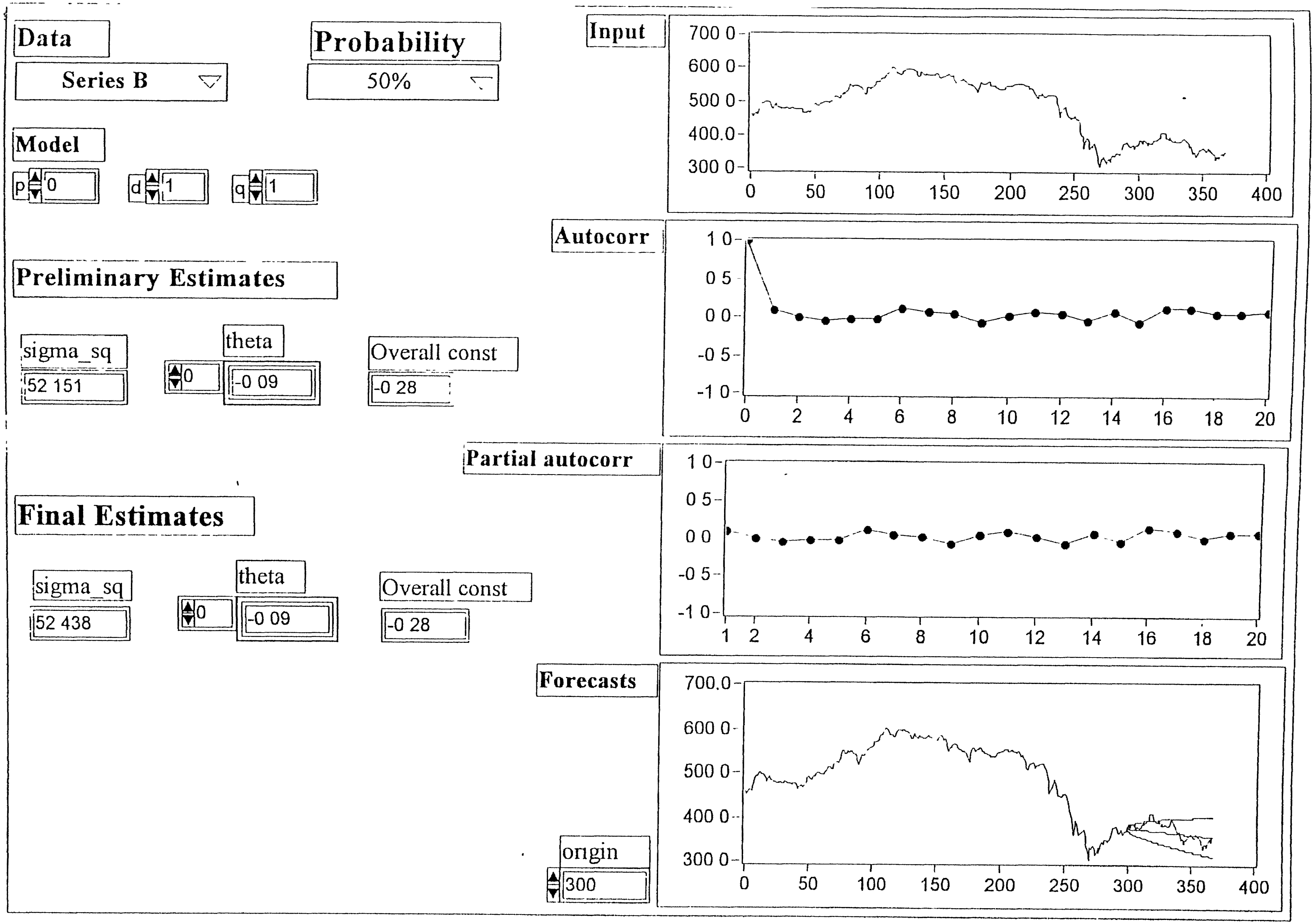


Fig. 3.4 Front Panel for series B

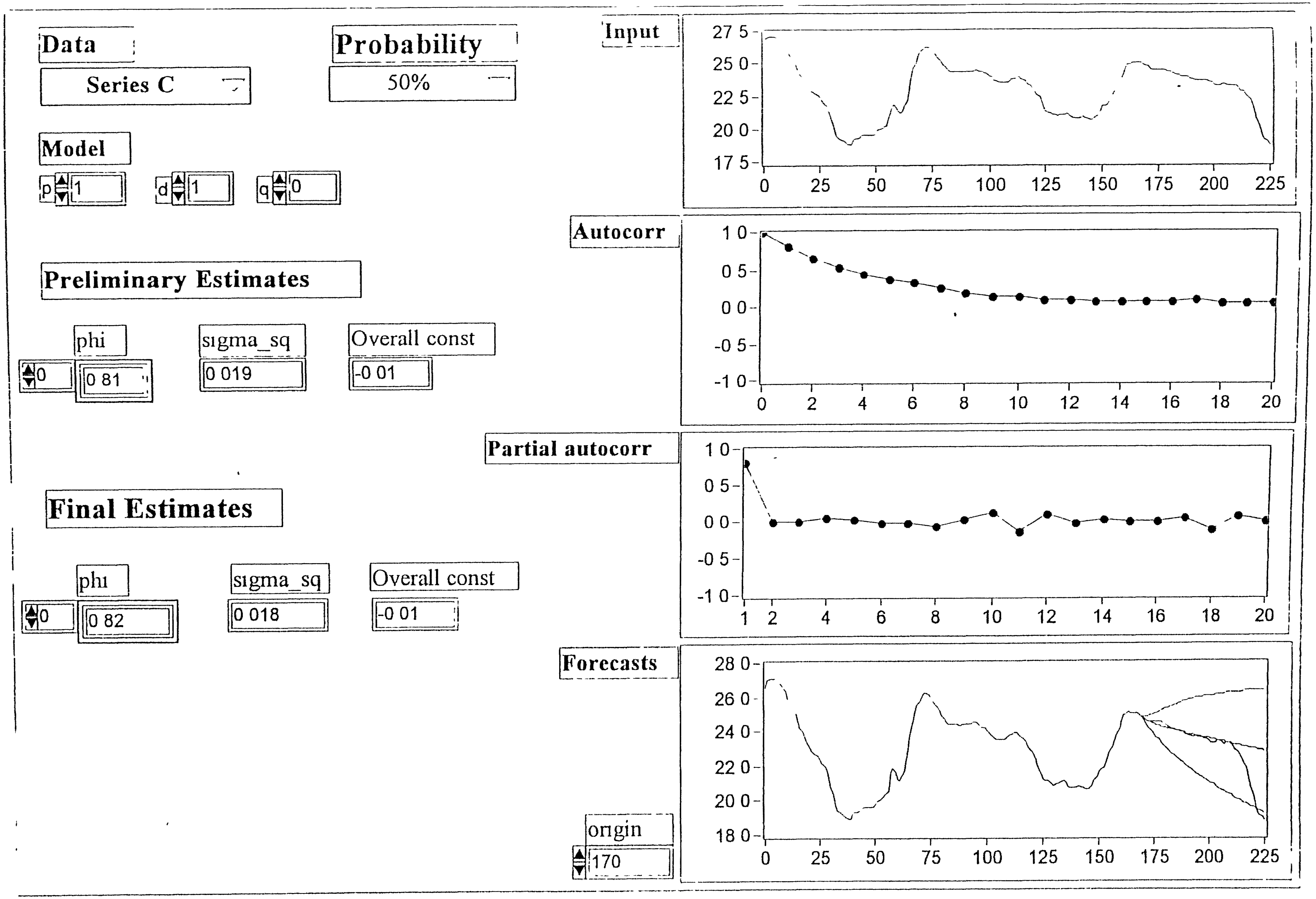


Fig. 3.5 Front Panel for series C

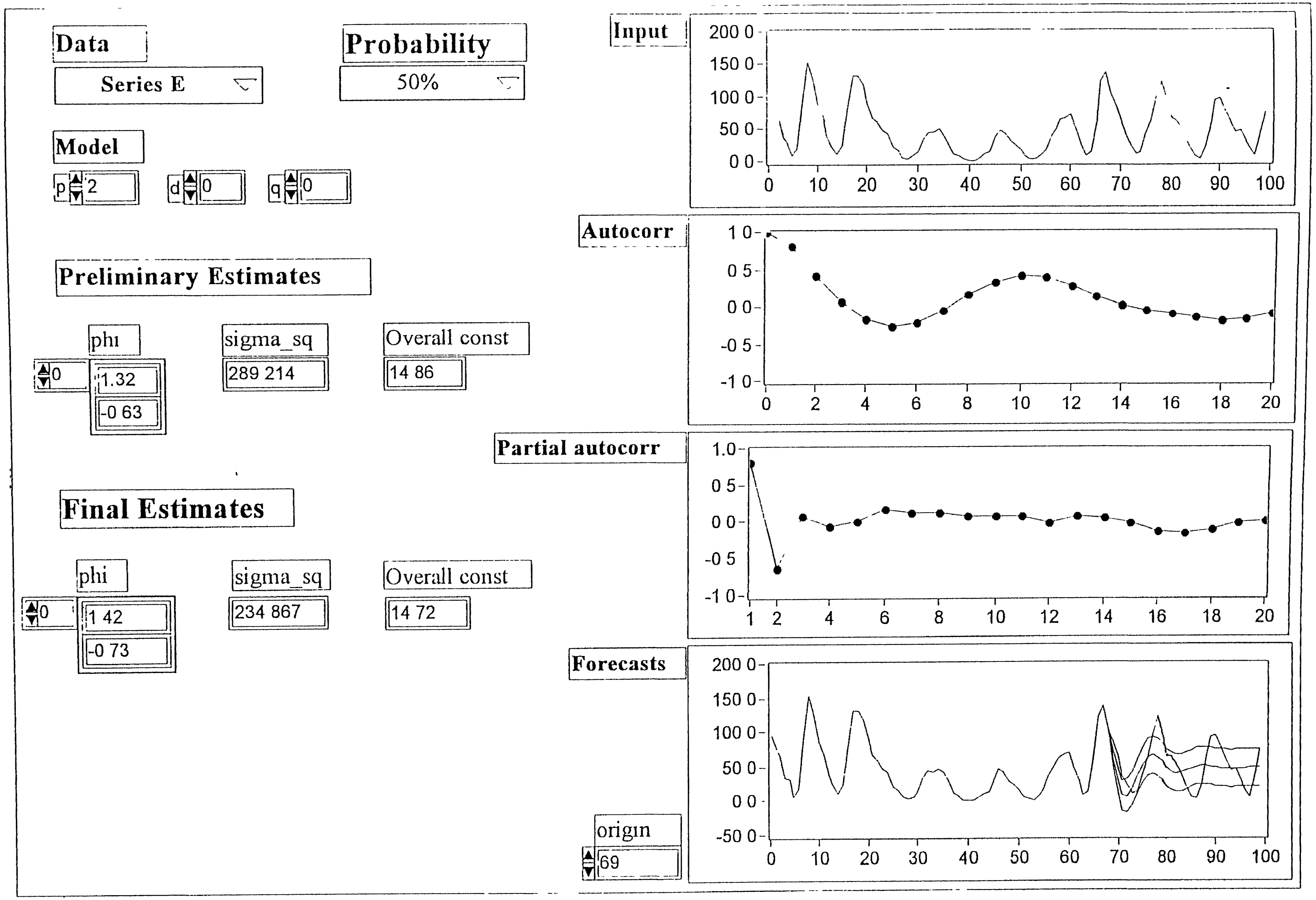


Fig. 3.7 Front Panel for series E

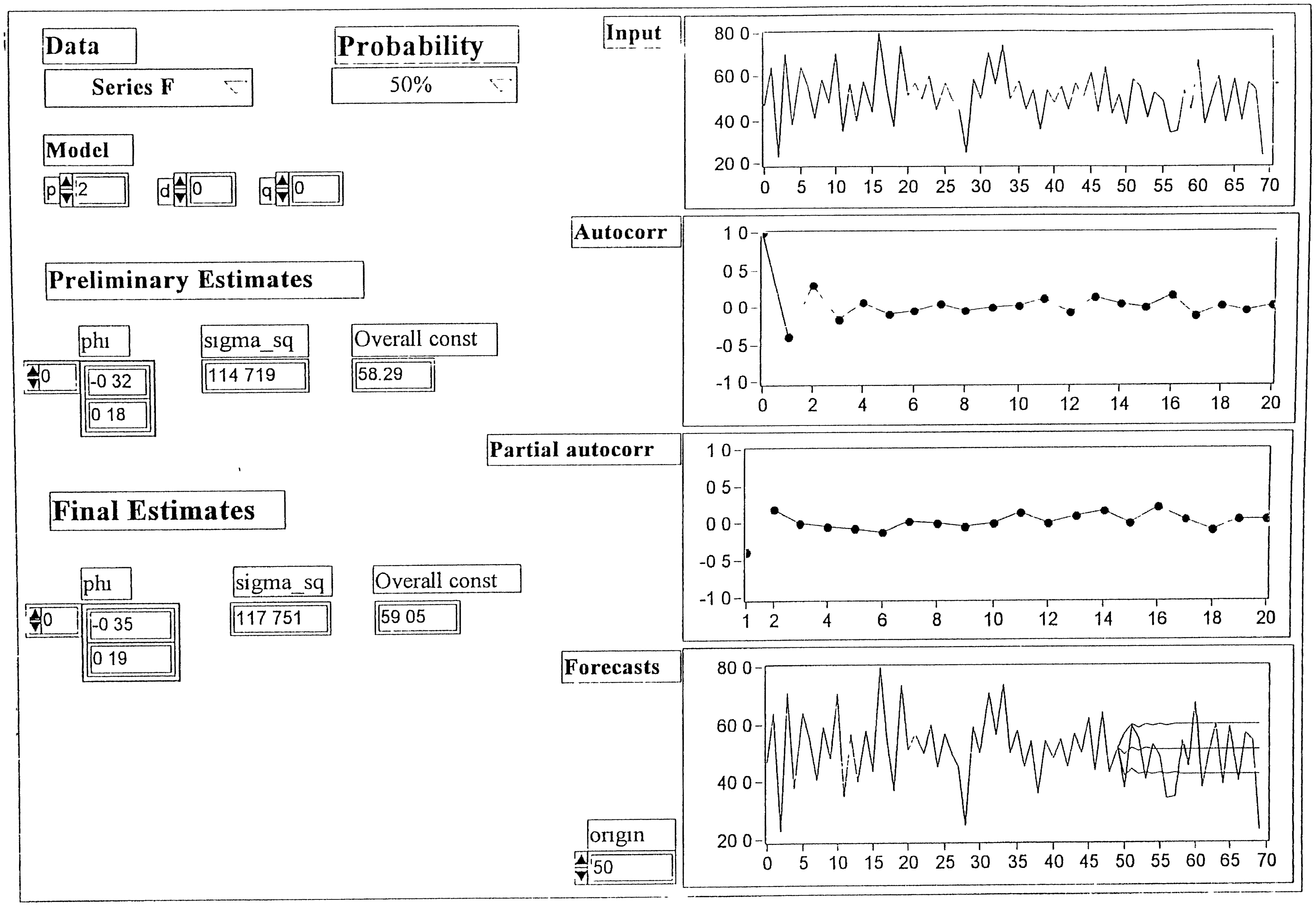


Fig. 3.8 Front Panel for series F

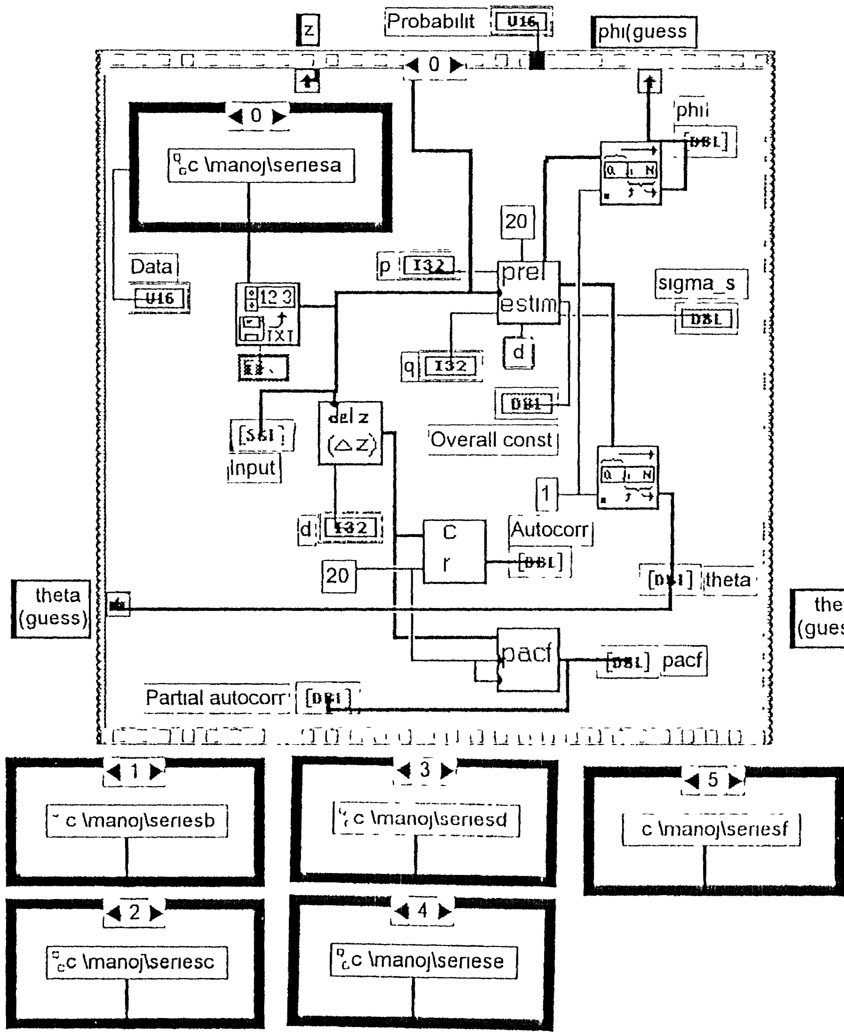


Fig. 3.9(a) Block Diagram for the front panel of Fig. 3.3

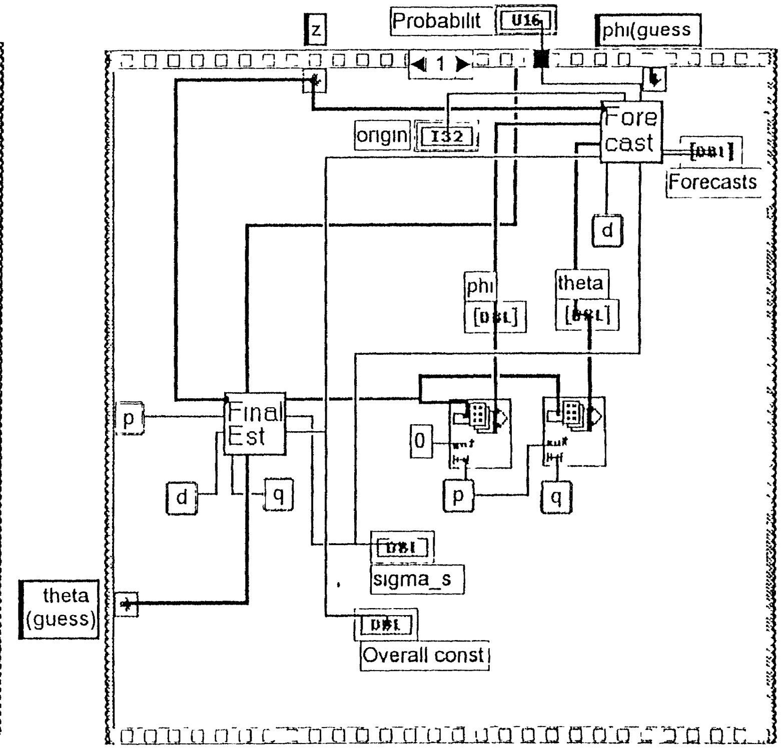


Fig. 3.9(b) Block Diagram for the front panel of Fig. 3.3

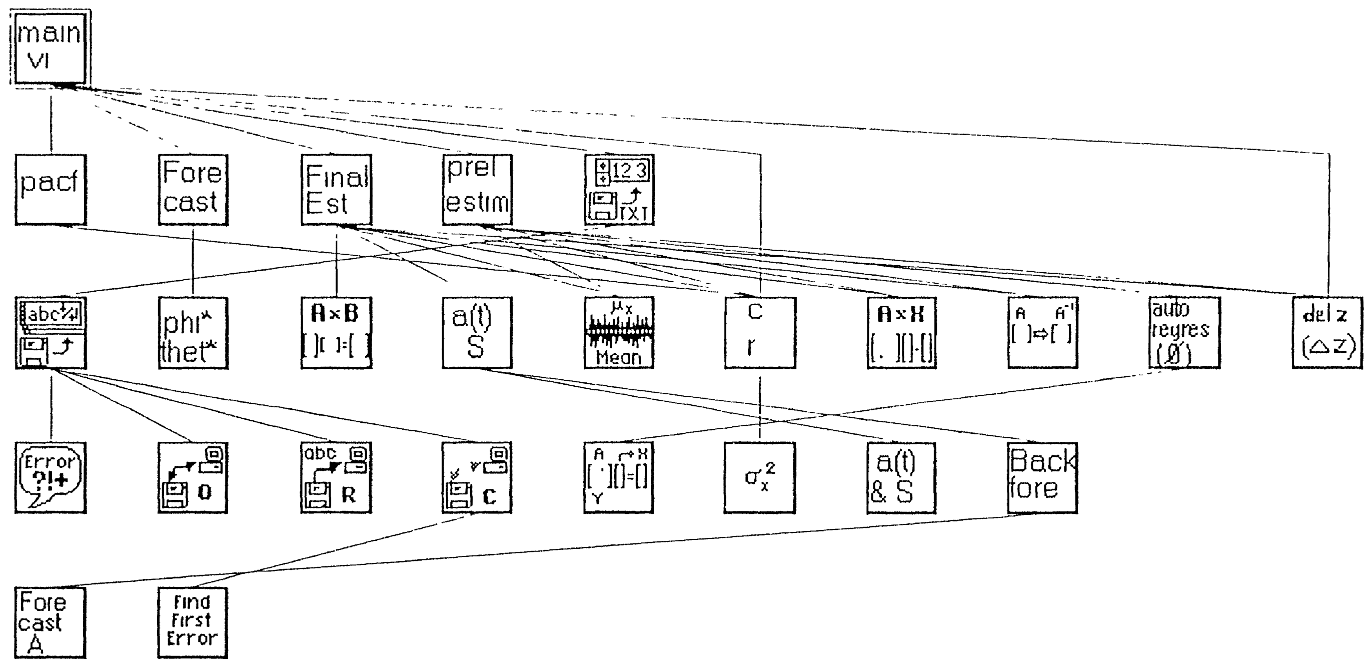
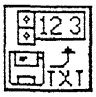
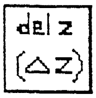


Fig. 3.10 Hierarchy of subVIs

List of SubVIs



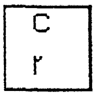
Read From Spreadsheet File.vi
C \LABVIEW\vi.lib\UTILITY\FILE LLB\Read From Spreadsheet File vi



del.vi
C:\LABVIEW\ARMA\THESIS.LLB\del.vi

prelim parameter estimation.vi
C:\LABVIEW\ARMA\THESIS LLB\prelim parameter estimation vi

Final estimation icon.vi
C.\LABVIEW\ARMA\THESIS.LLB\Final estimation icon.vi



c & r.vi
C:\LABVIEW\ARMA\THESIS LLB\c & r.vi

forecast.vi
C.\LABVIEW\ARMA\THESIS LLB\forecast.vi

pacf.vi
C \LABVIEW\ARMA\THESIS LLB\pacf vi

Fig. 3.11 Consolidated list of subVIs

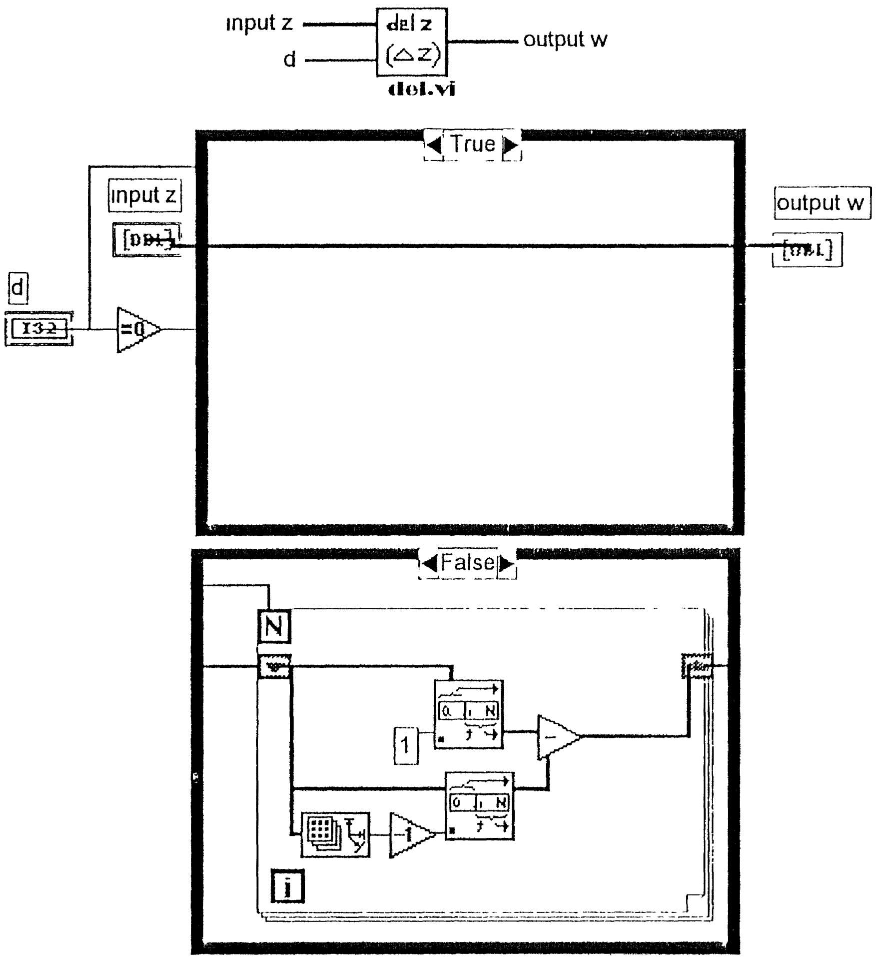


Fig. 3.12 VI for calculating difference

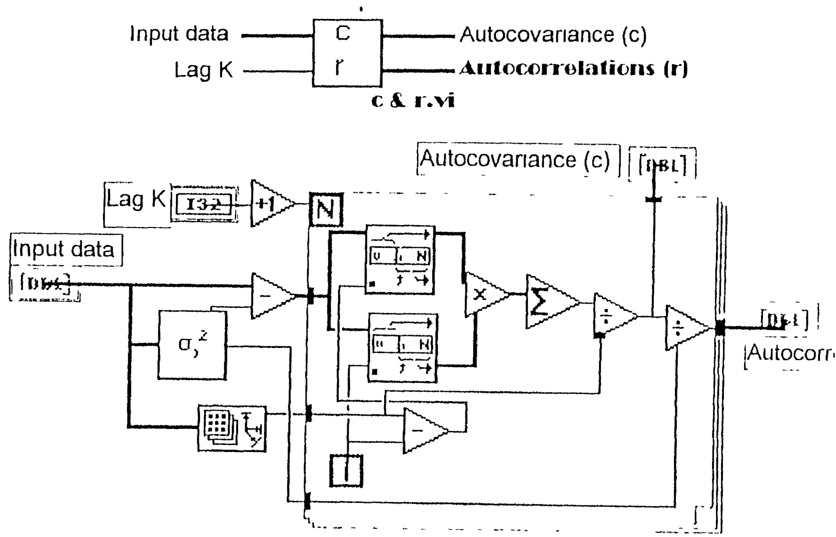


Fig. 3.13 VI for calculating autocorrelation

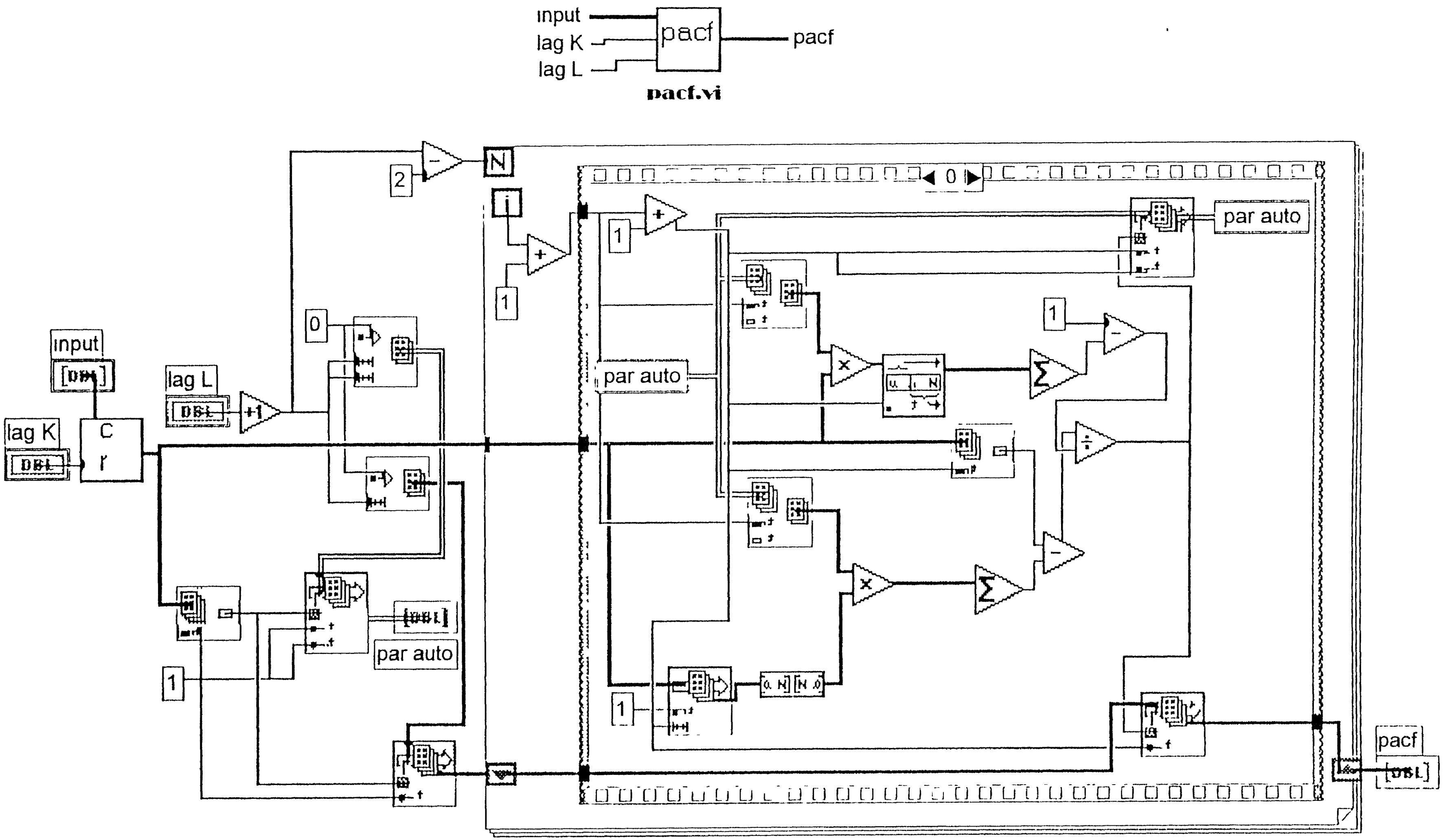


Fig. 3.14(a) VI for calculating partial autocorrelation

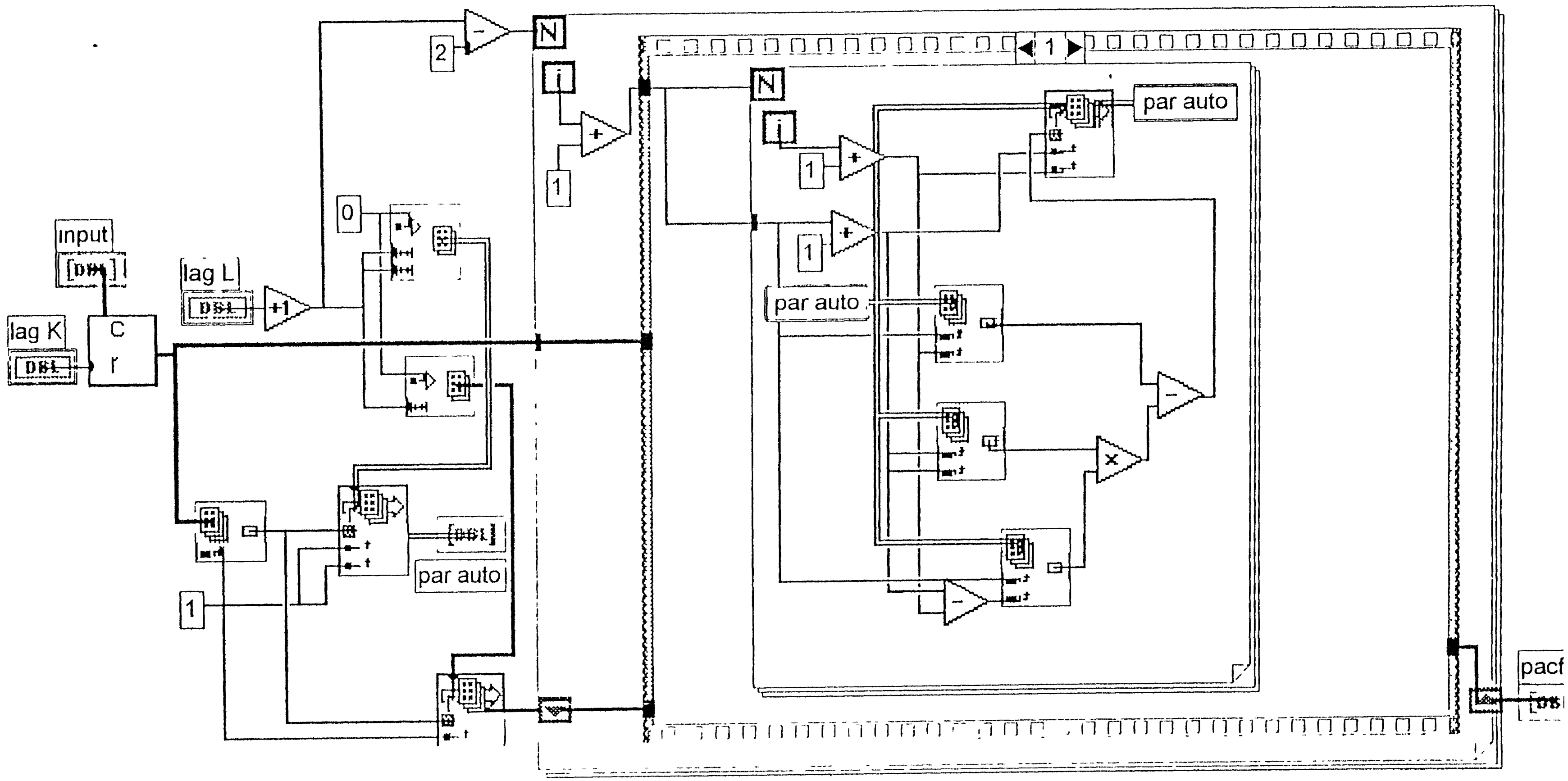
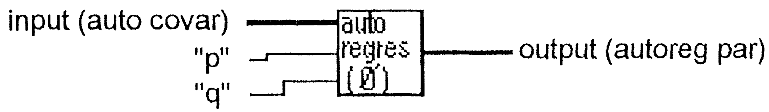


Fig. 3.14(b) VI for calculating partial autocorrelation



autoreg_para.vi

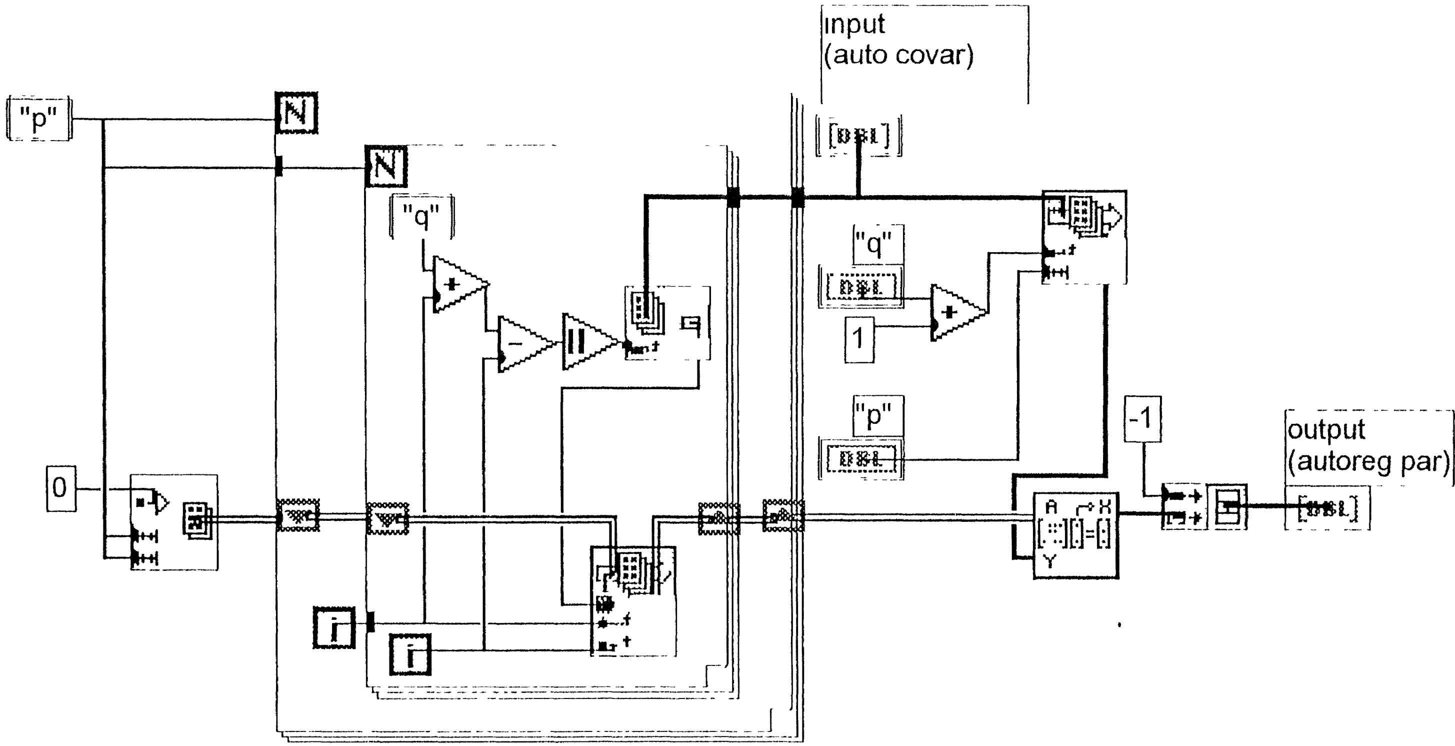


Fig. 3.15 VI for calculating initial estimates of autoregressive parameters

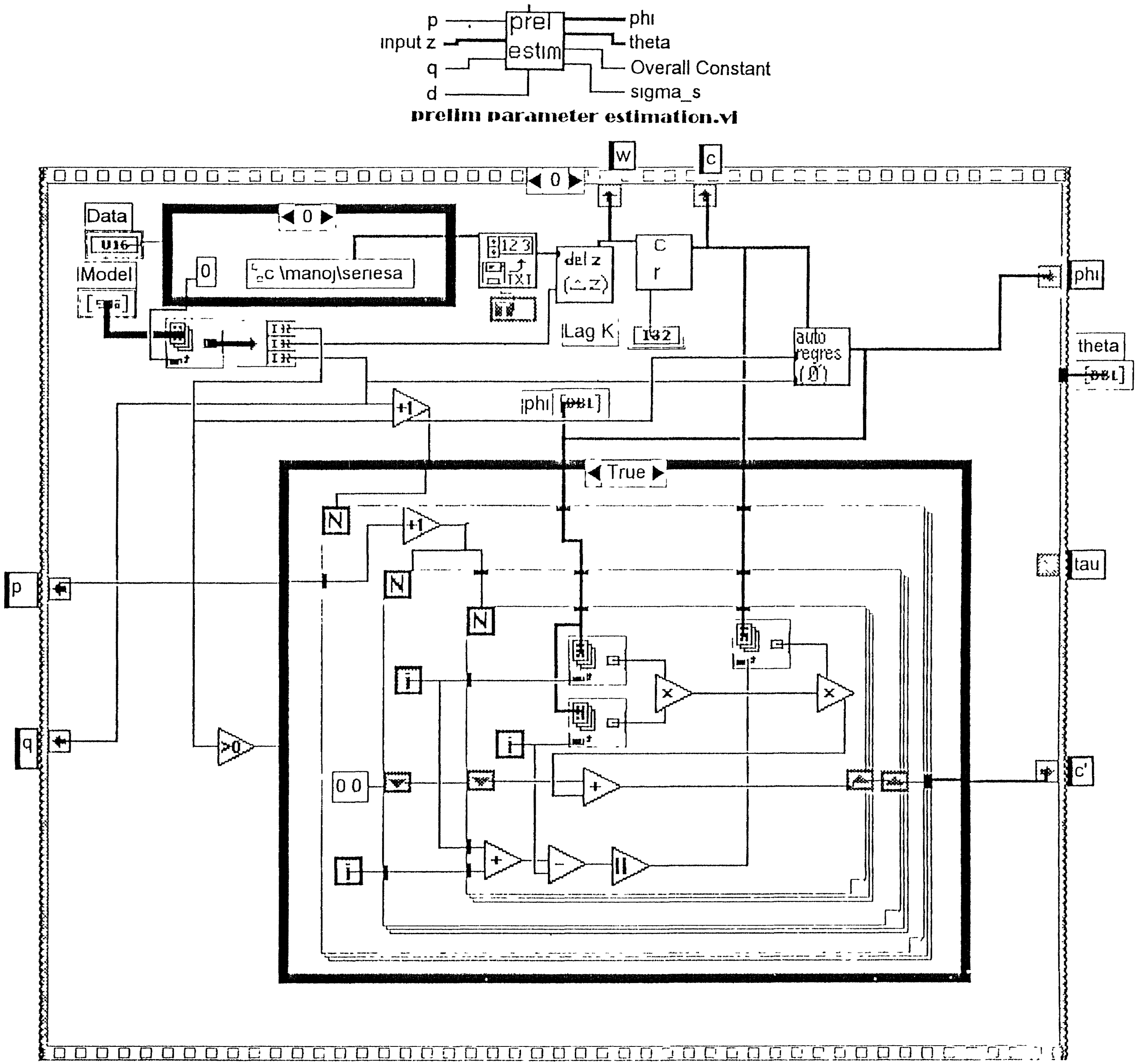


Fig. 3.16(a) VI for calculating the initial estimates of a general ARMA Model

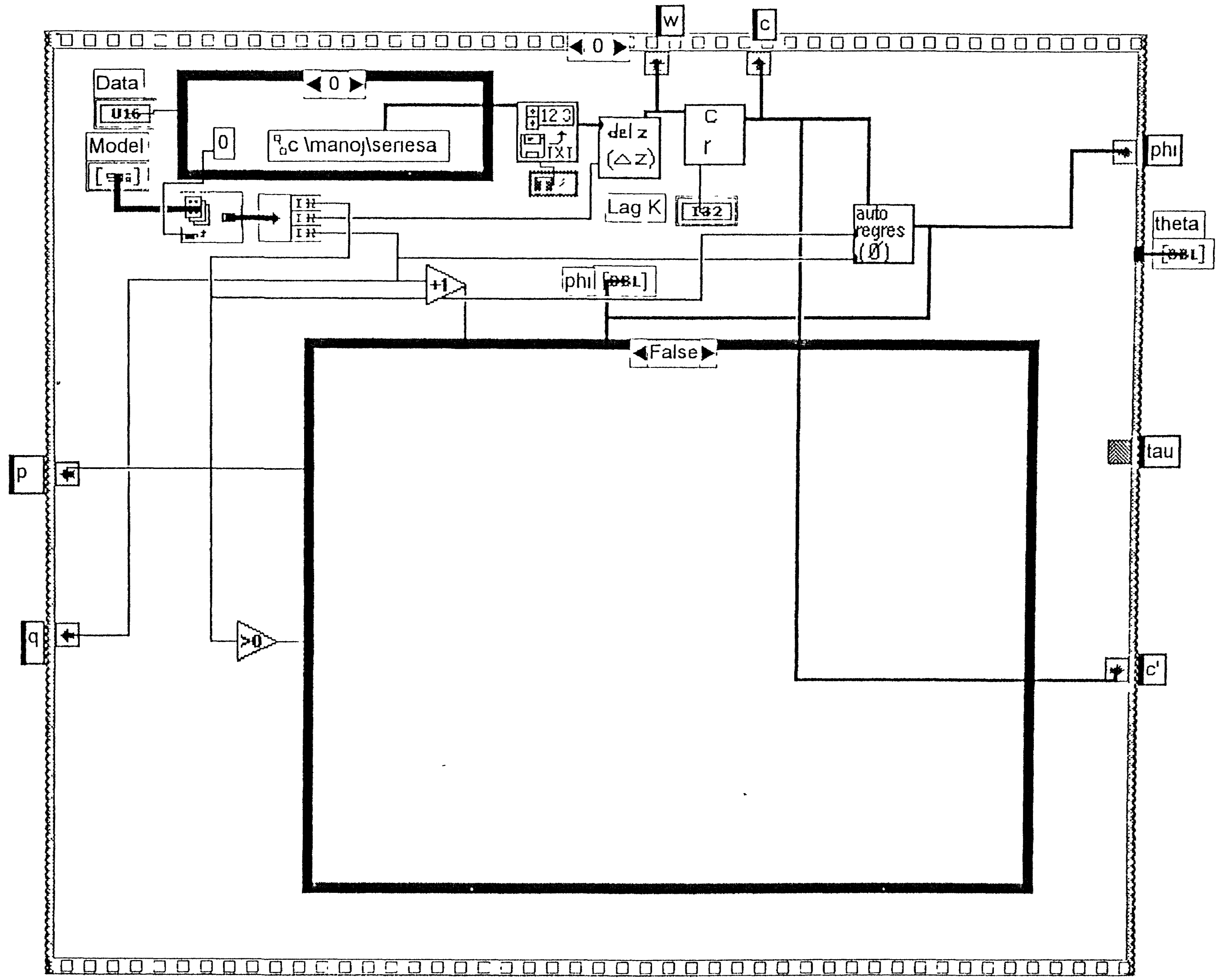


Fig. 3.16(b) VI for calculating the initial estimates of a general ARMA Model

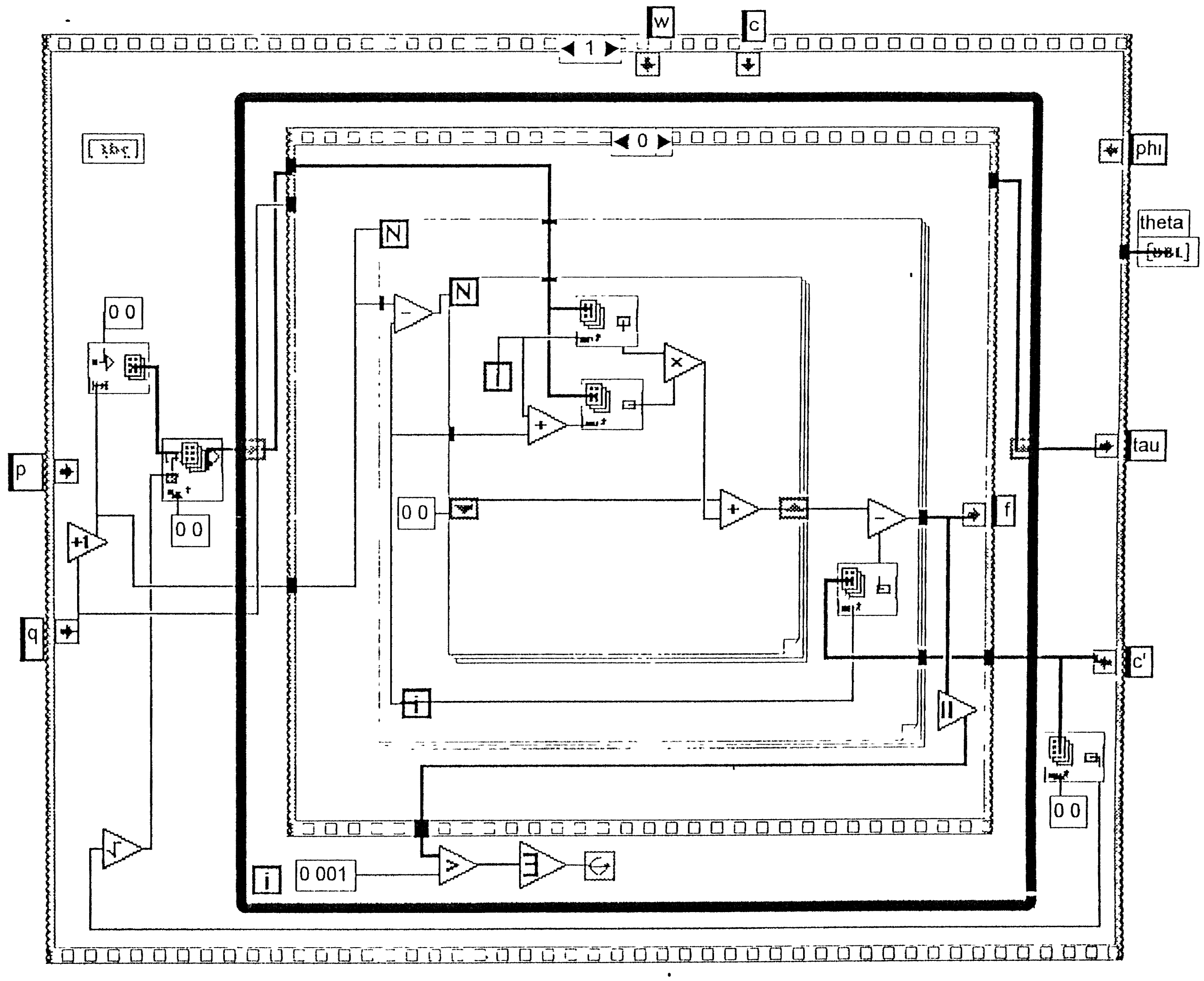


Fig. 3.16(c) VI for calculating the initial estimates of a general ARMA Model

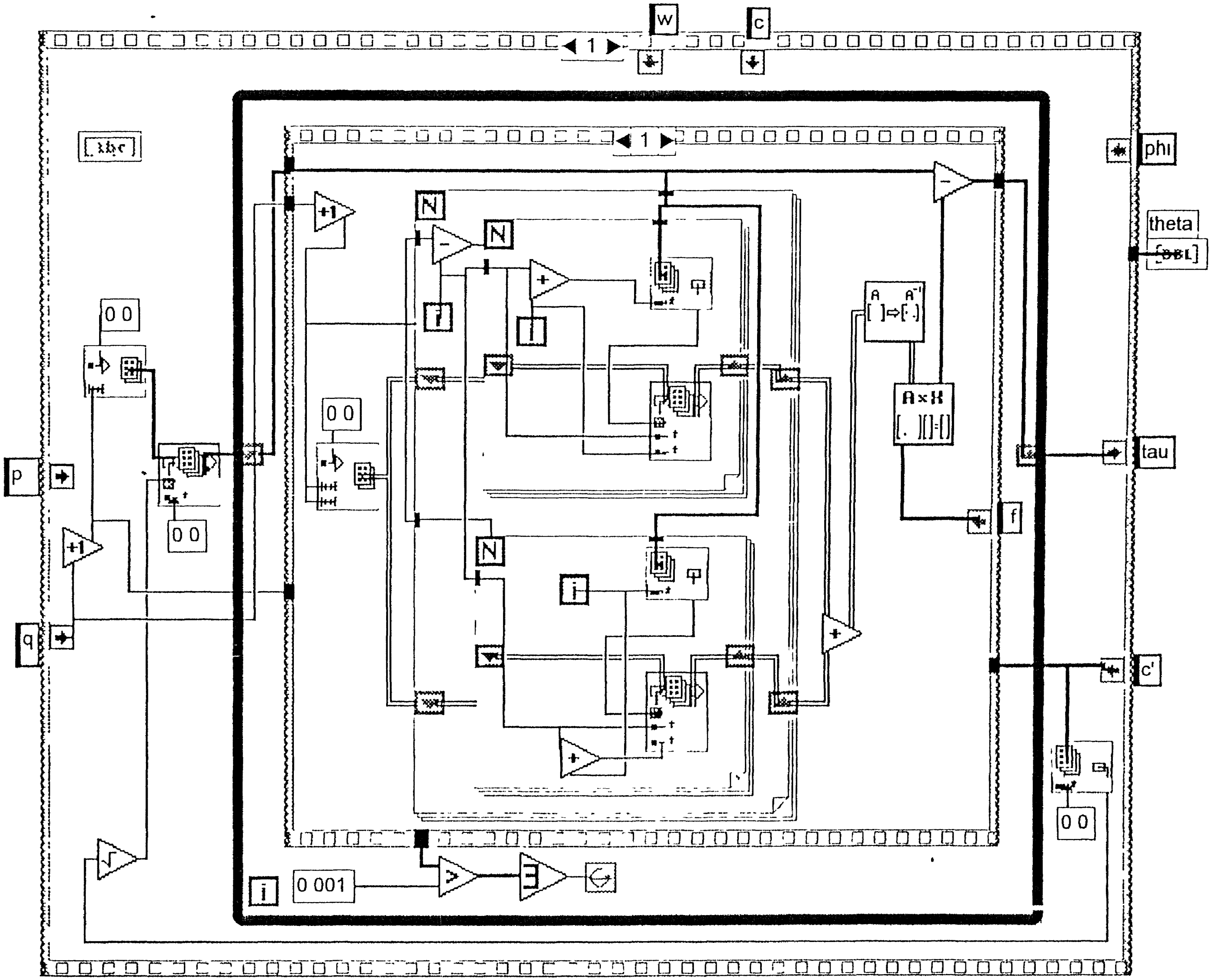


Fig. 3.16(d) VI for calculating the initial estimates of a general ARMA Model

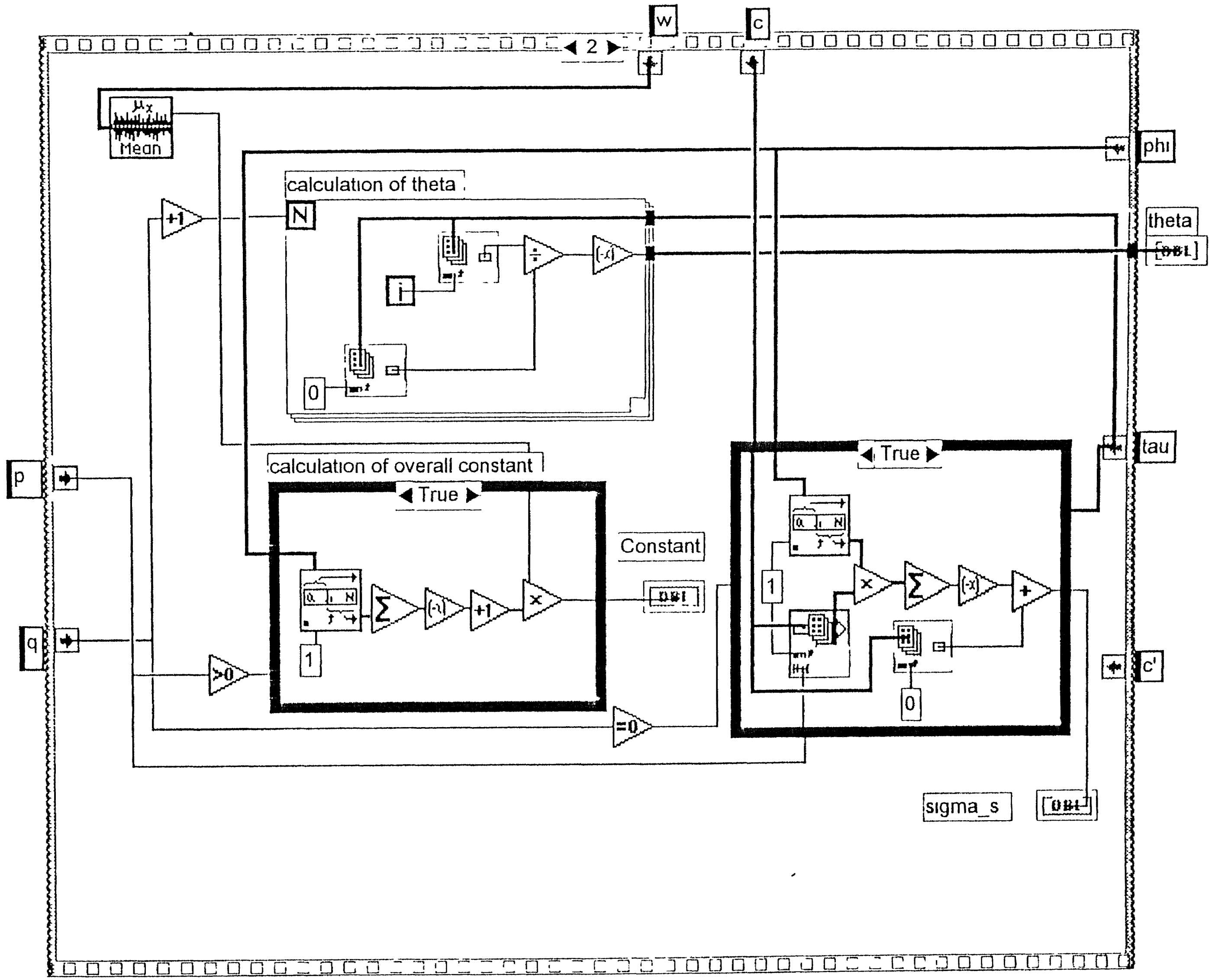


Fig. 3.16(e) VI for calculating the initial estimates of a general ARMA Model

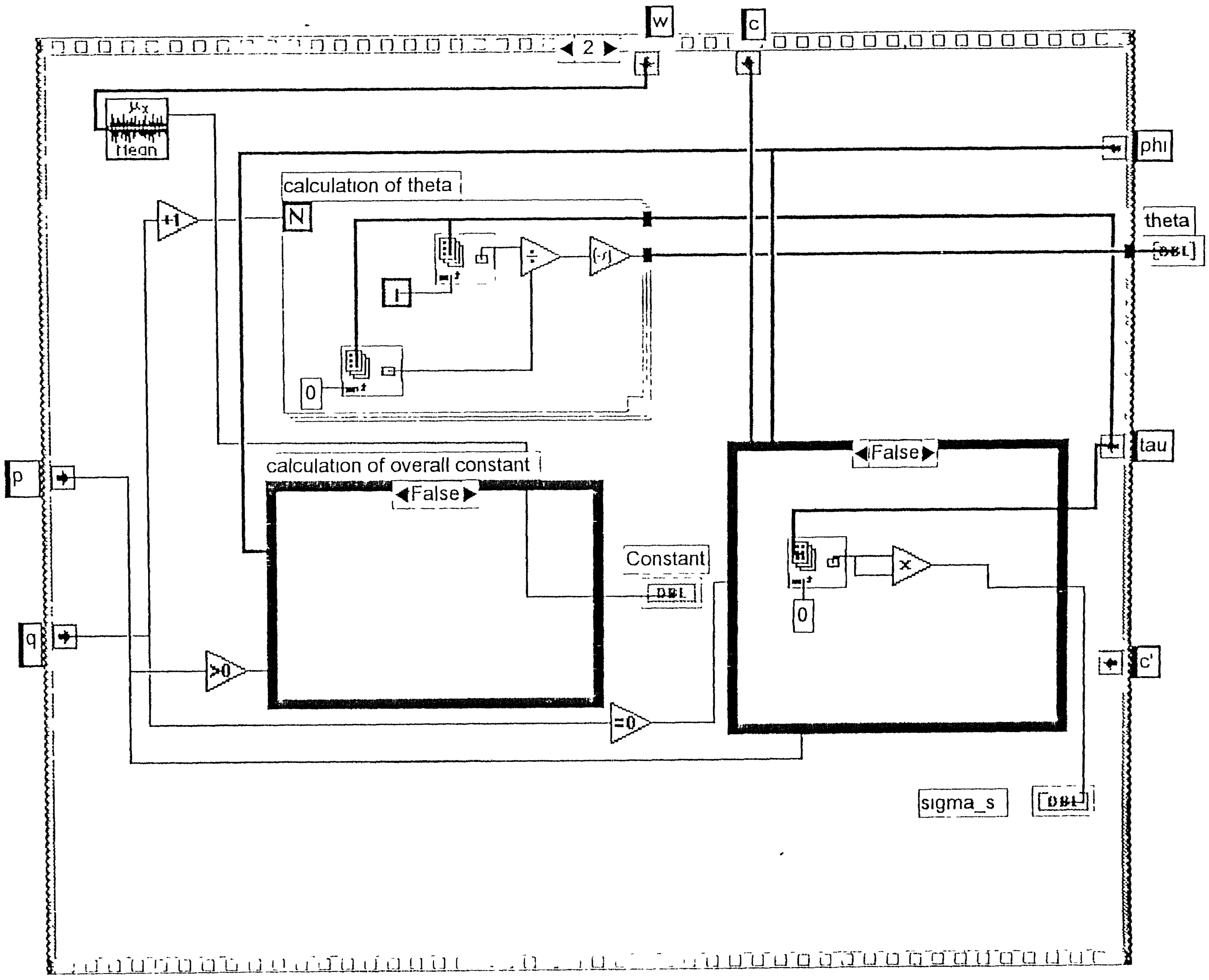


Fig. 3.16(f) VI for calculating the initial estimates of a general ARMA Model

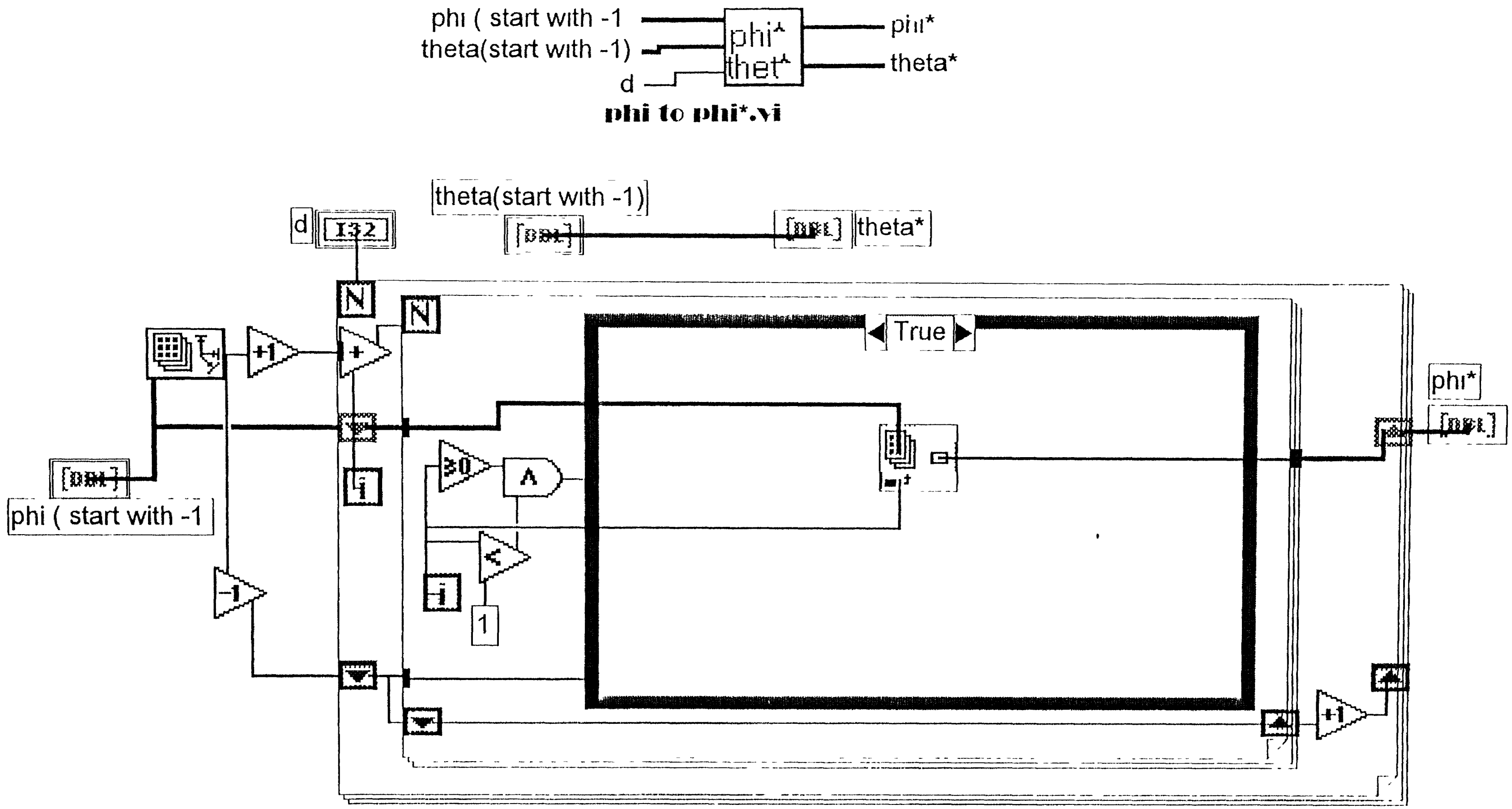


Fig. 3.17(a) VI to convert ϕ to φ .

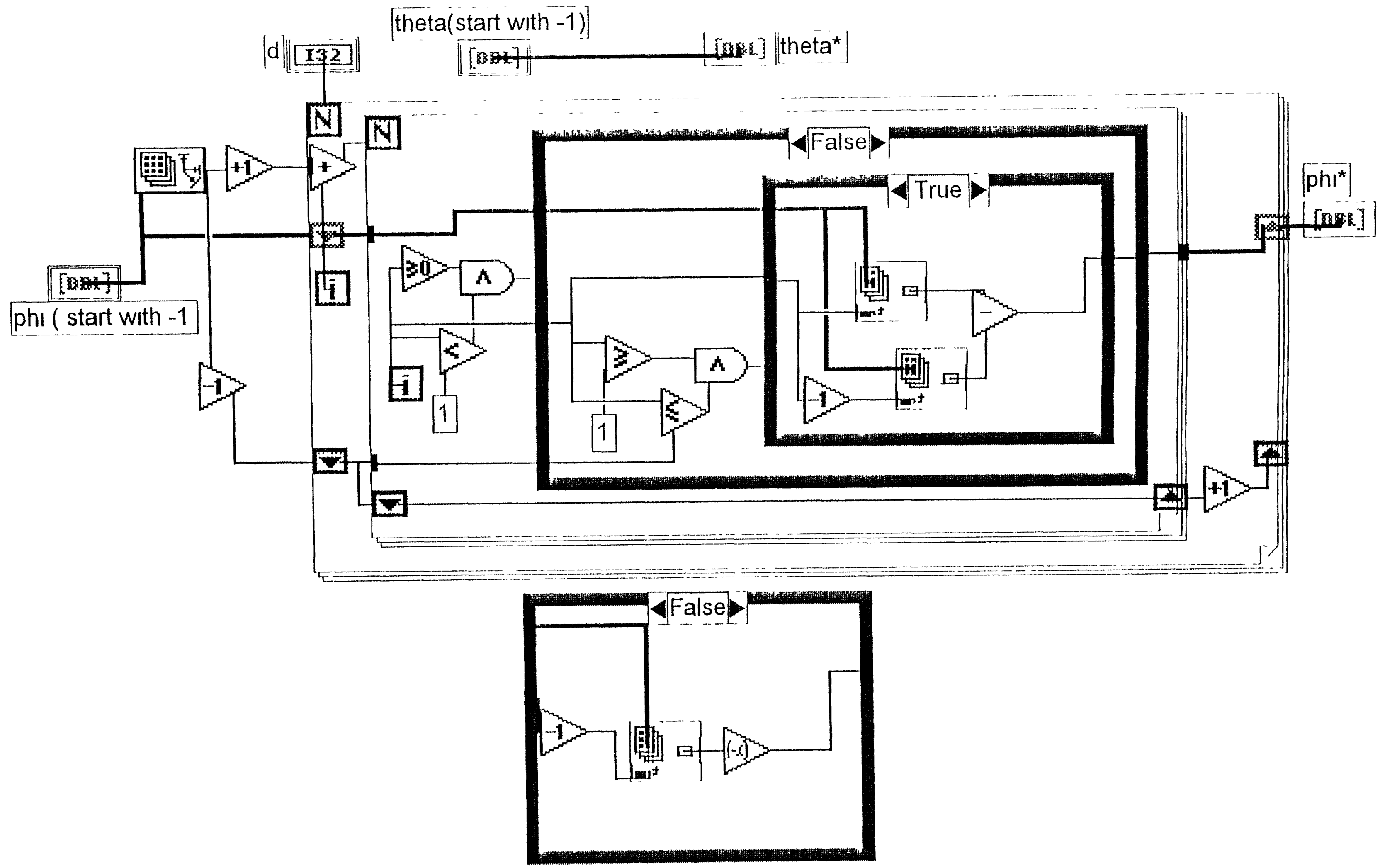
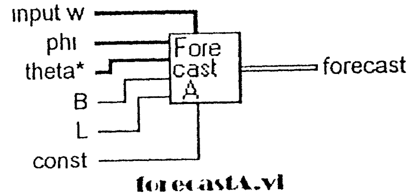


Fig. 3.17(b) VI to convert ϕ to φ



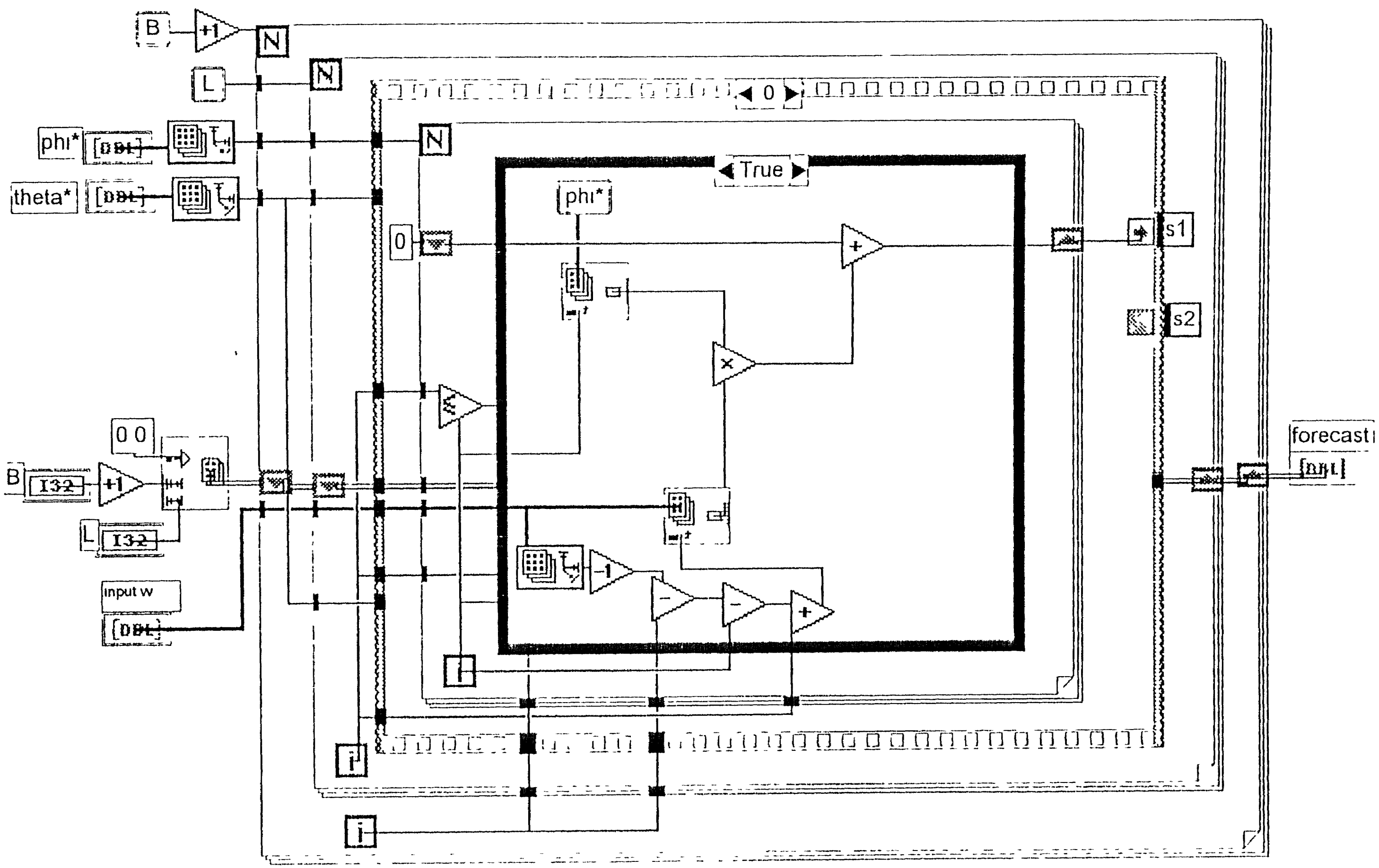


Fig. 3.18(a) VI for calculating forecasts at origin t for lead l

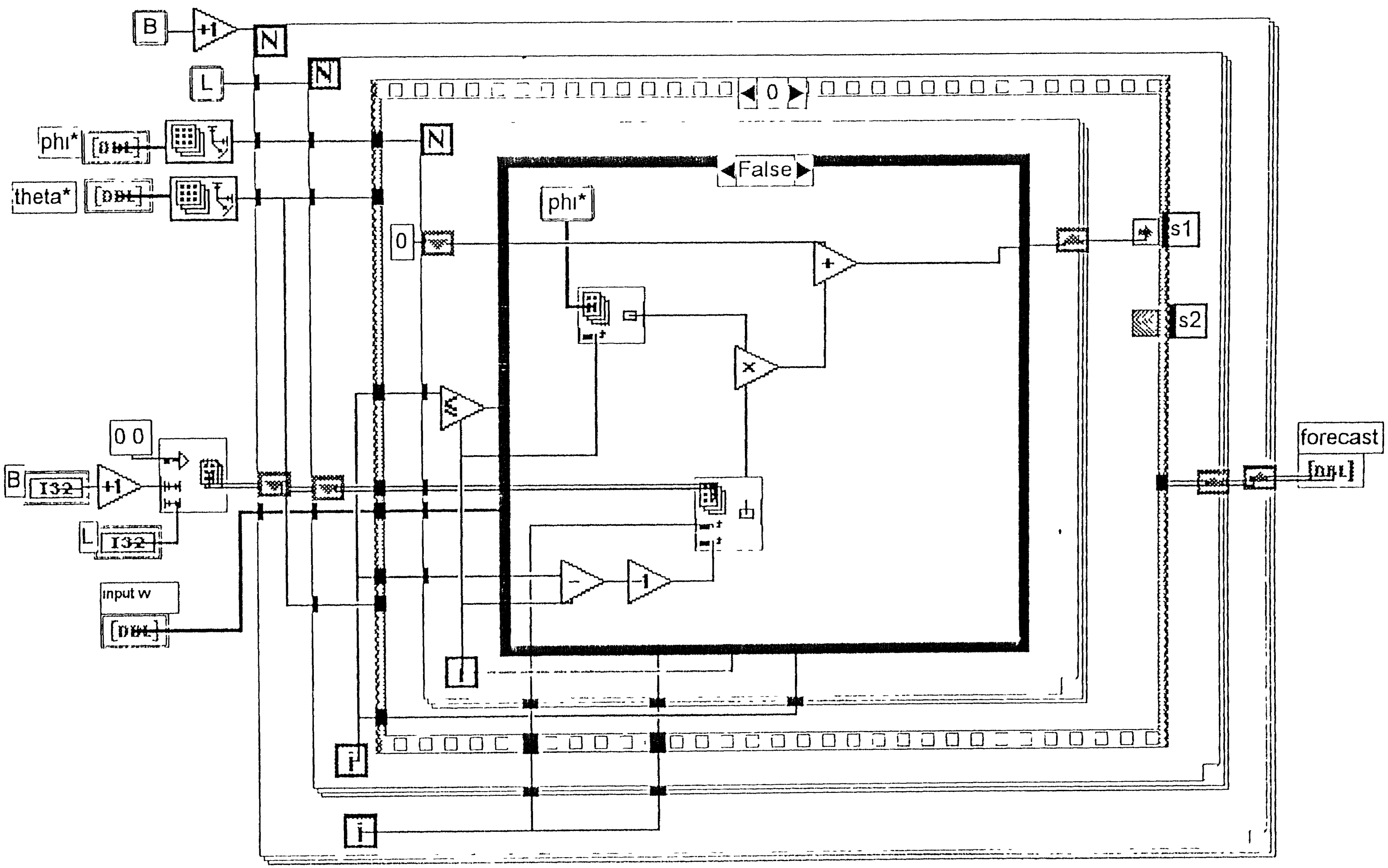


Fig. 3.18(b) VI for calculating forecasts at origin *t* for lead *l*

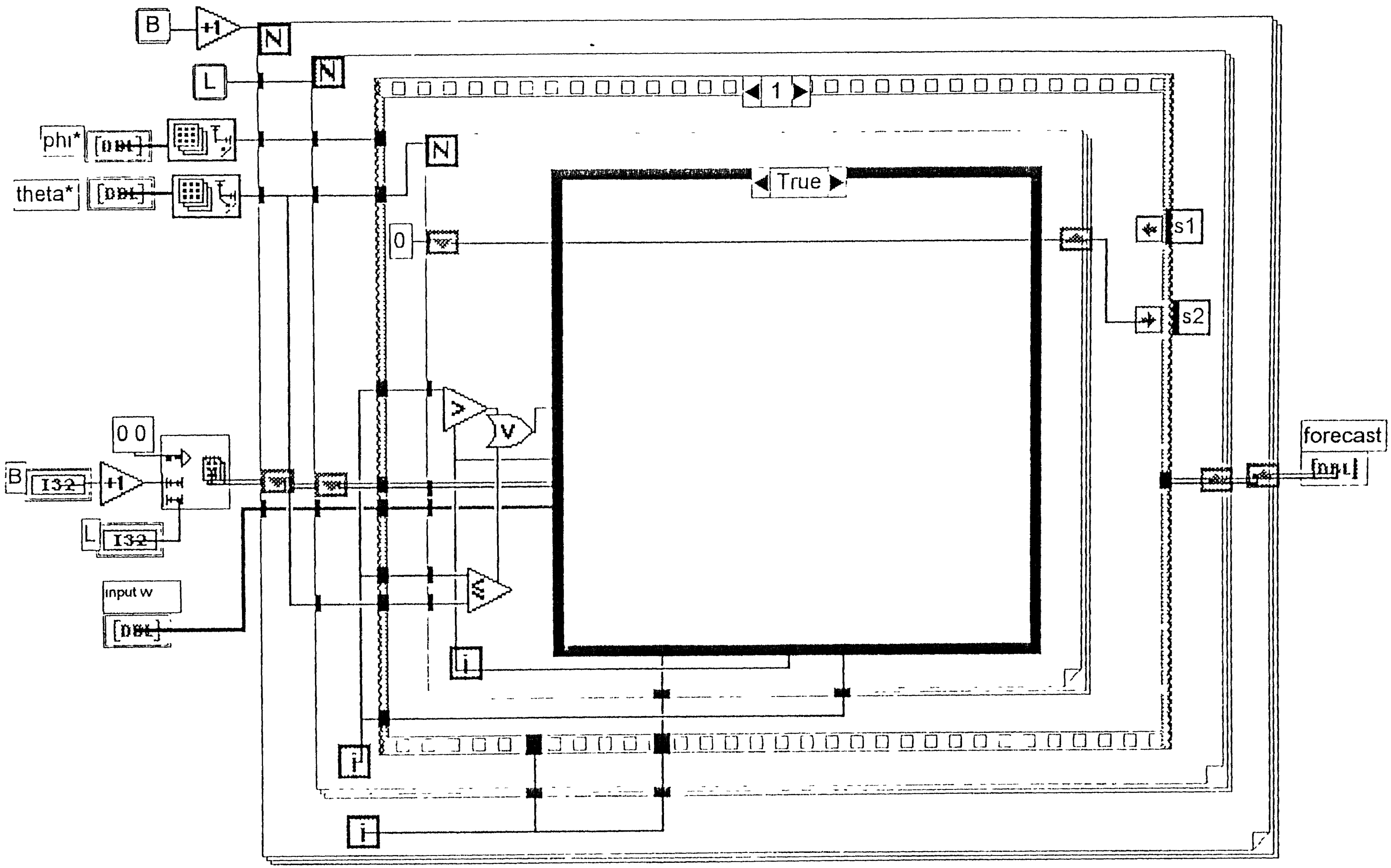


Fig. 3.18(c) VI for calculating forecasts at origin *t* for lead *l*

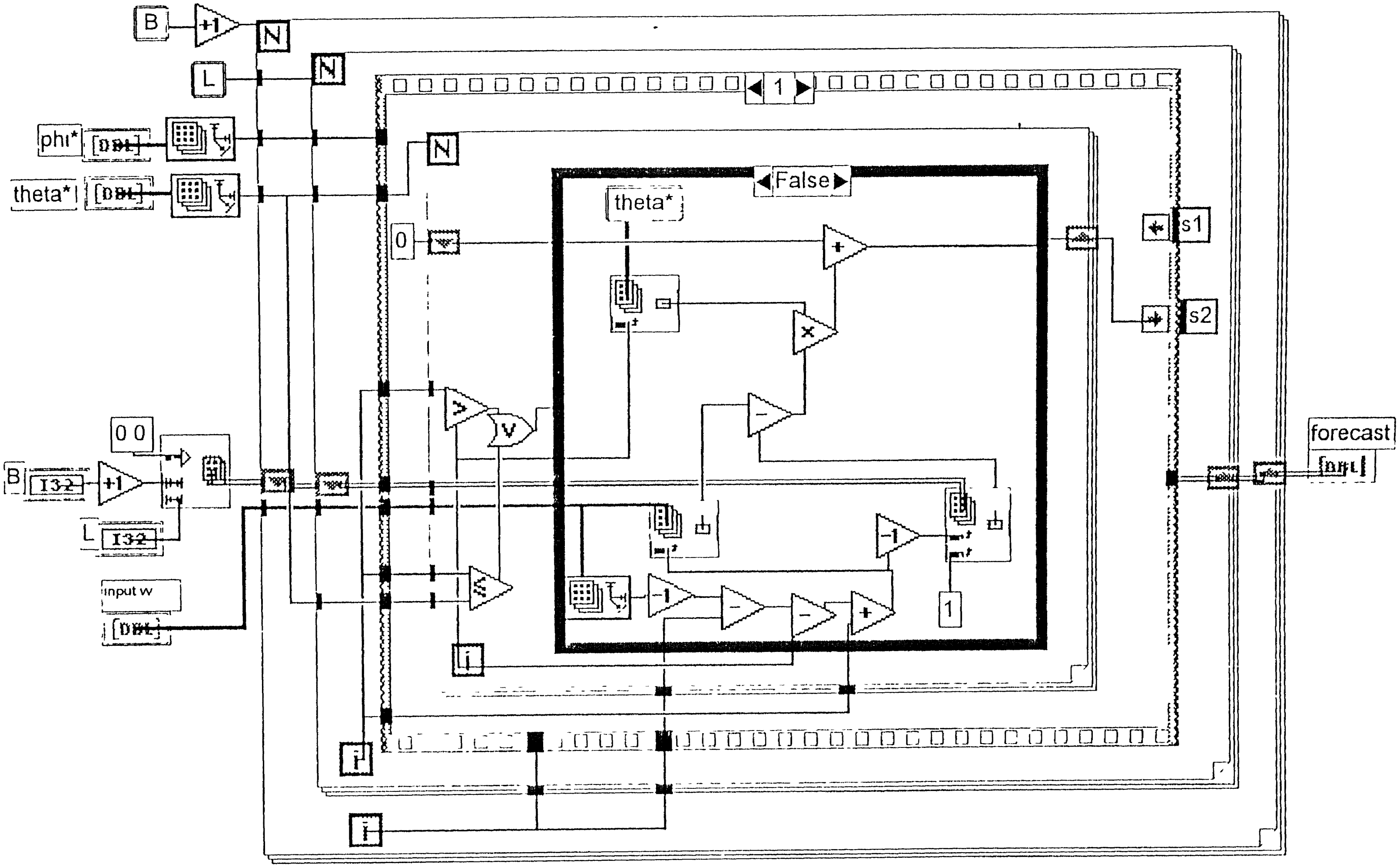


Fig. 3.18(d) VI for calculating forecasts at origin *t* for lead *l*

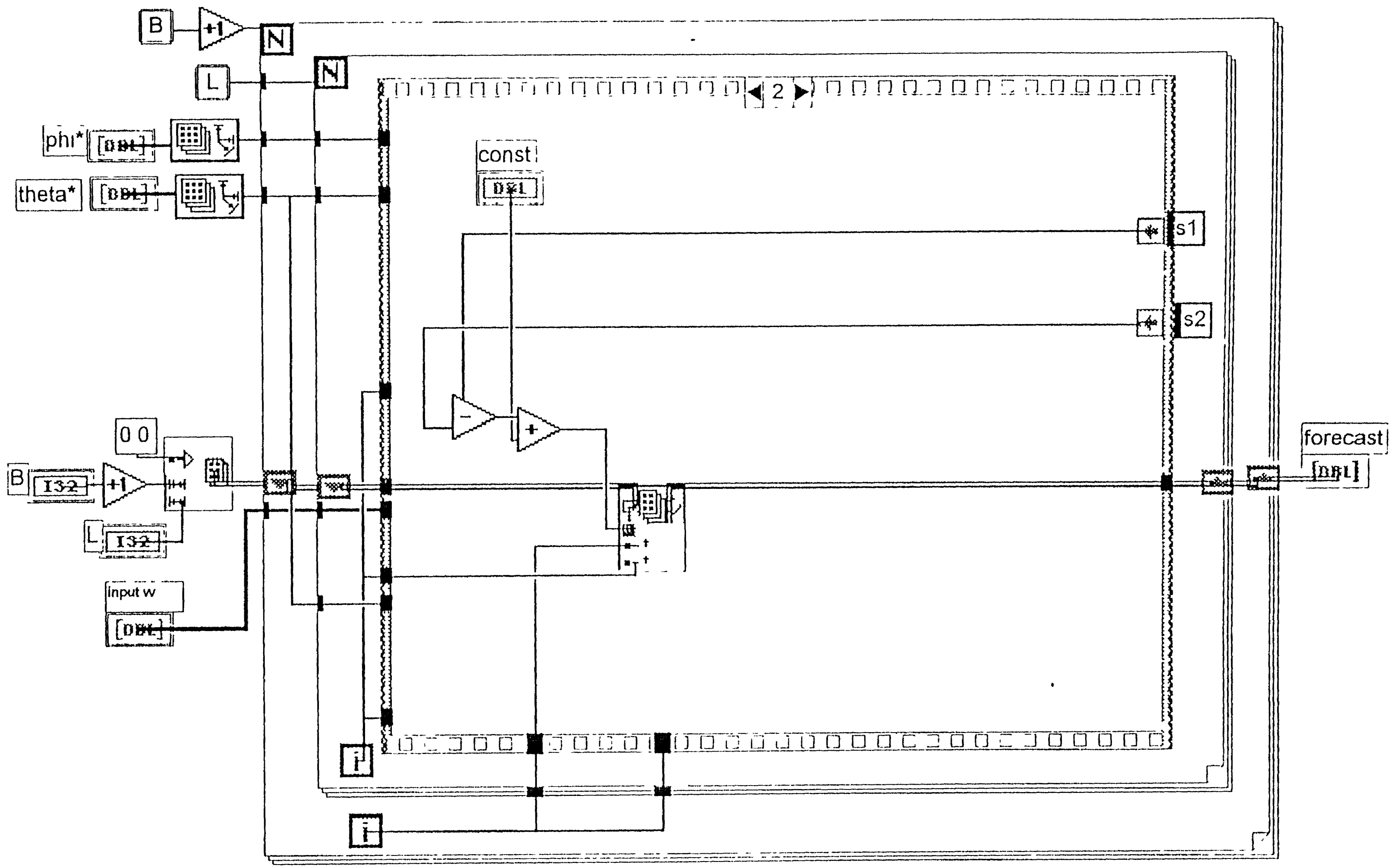
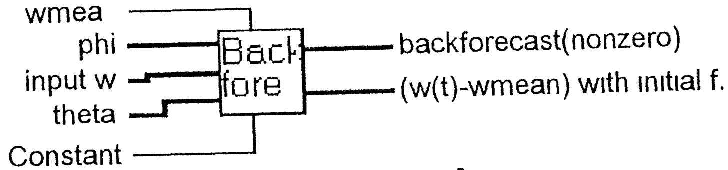


Fig. 3.18(e) VI for calculating forecasts at origin t for lead l



back_fore.vi

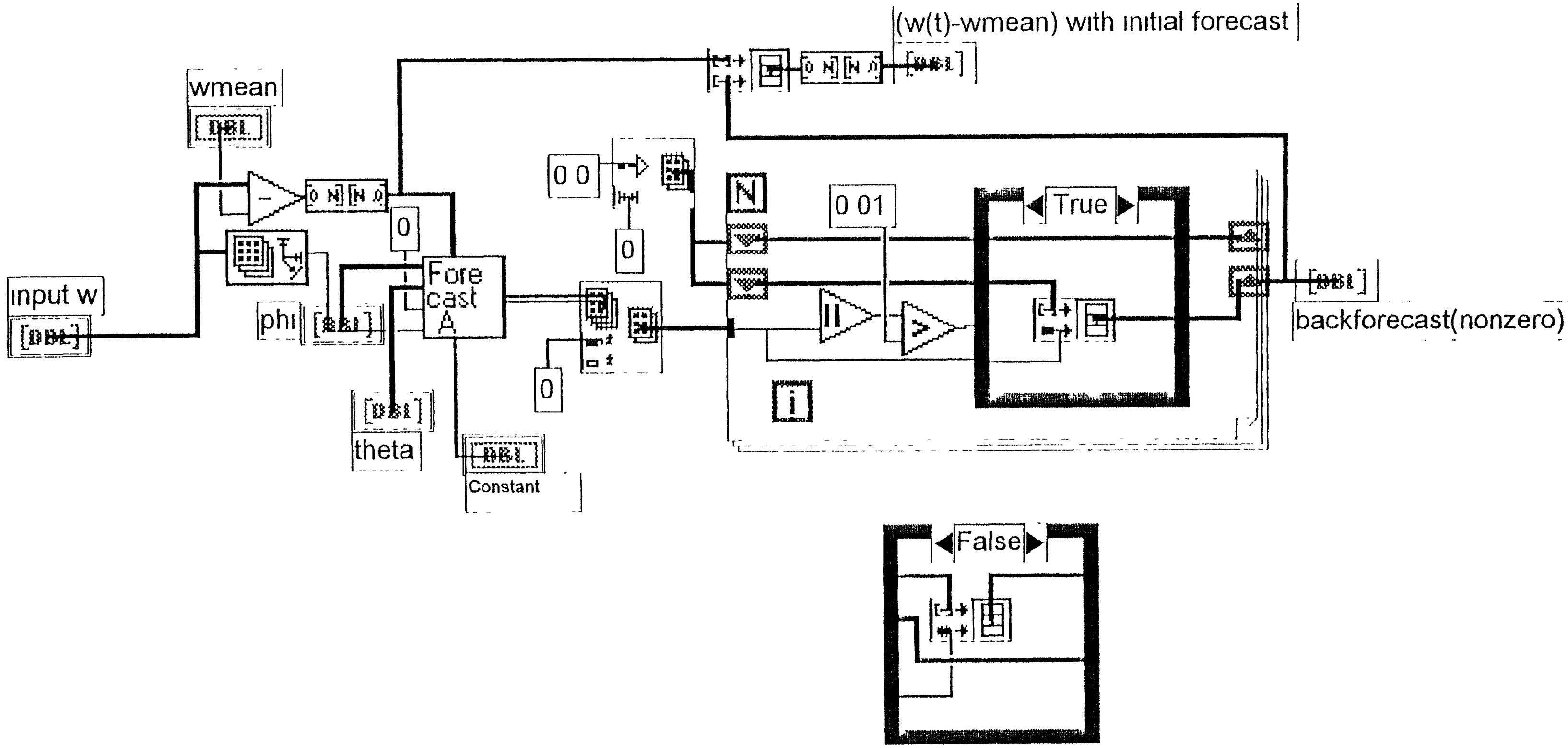


Fig. 3.19 VI for calculating back forecasts for $(w-\overline{w})$ series

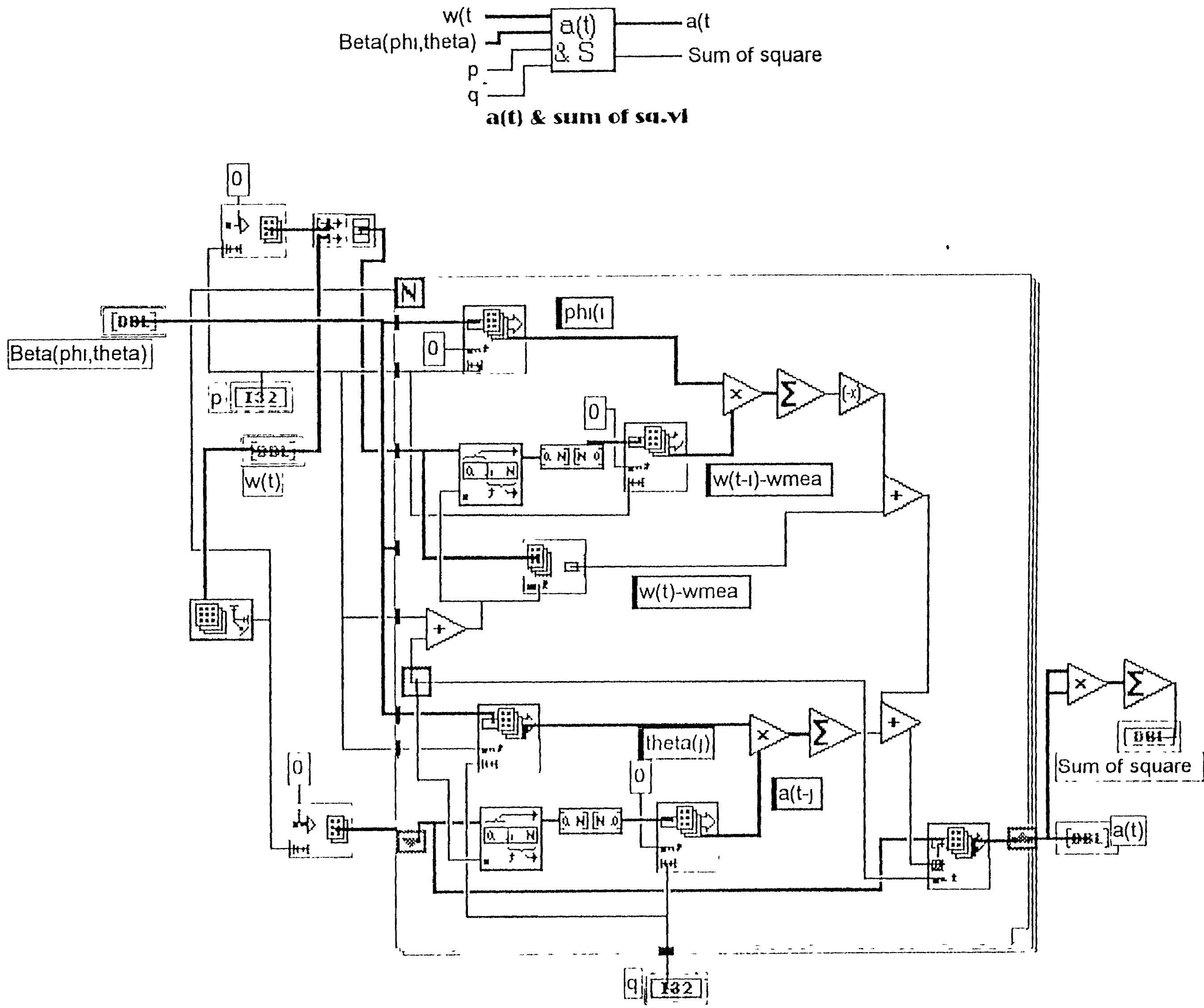


Fig. 3.20 VI for calculating a_t and S

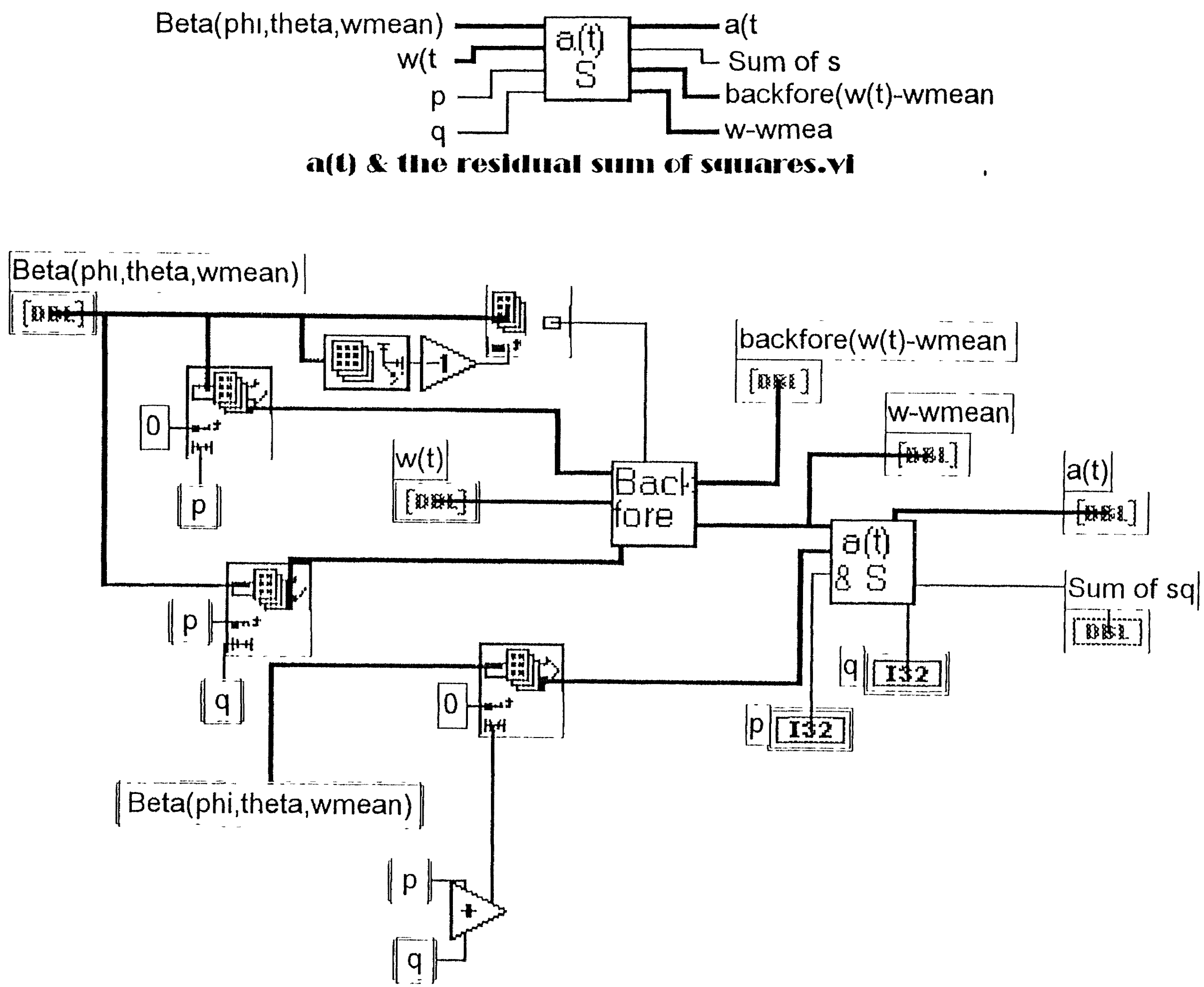


Fig. 3.21 **VI for calculating a_t and S using fig 3.18 and 3.19**

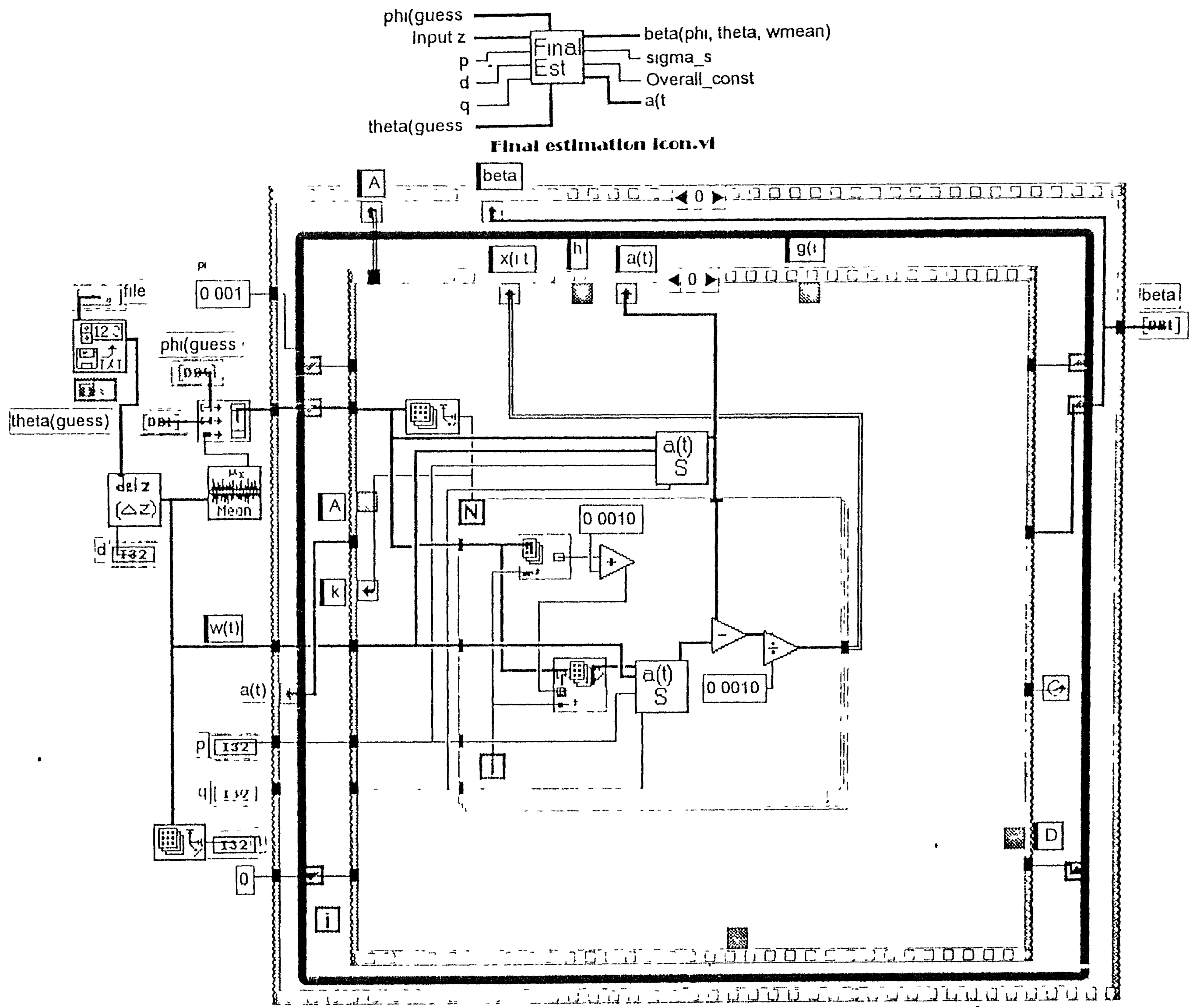


Fig. 3.22(a) VI for calculating final parameter estimation

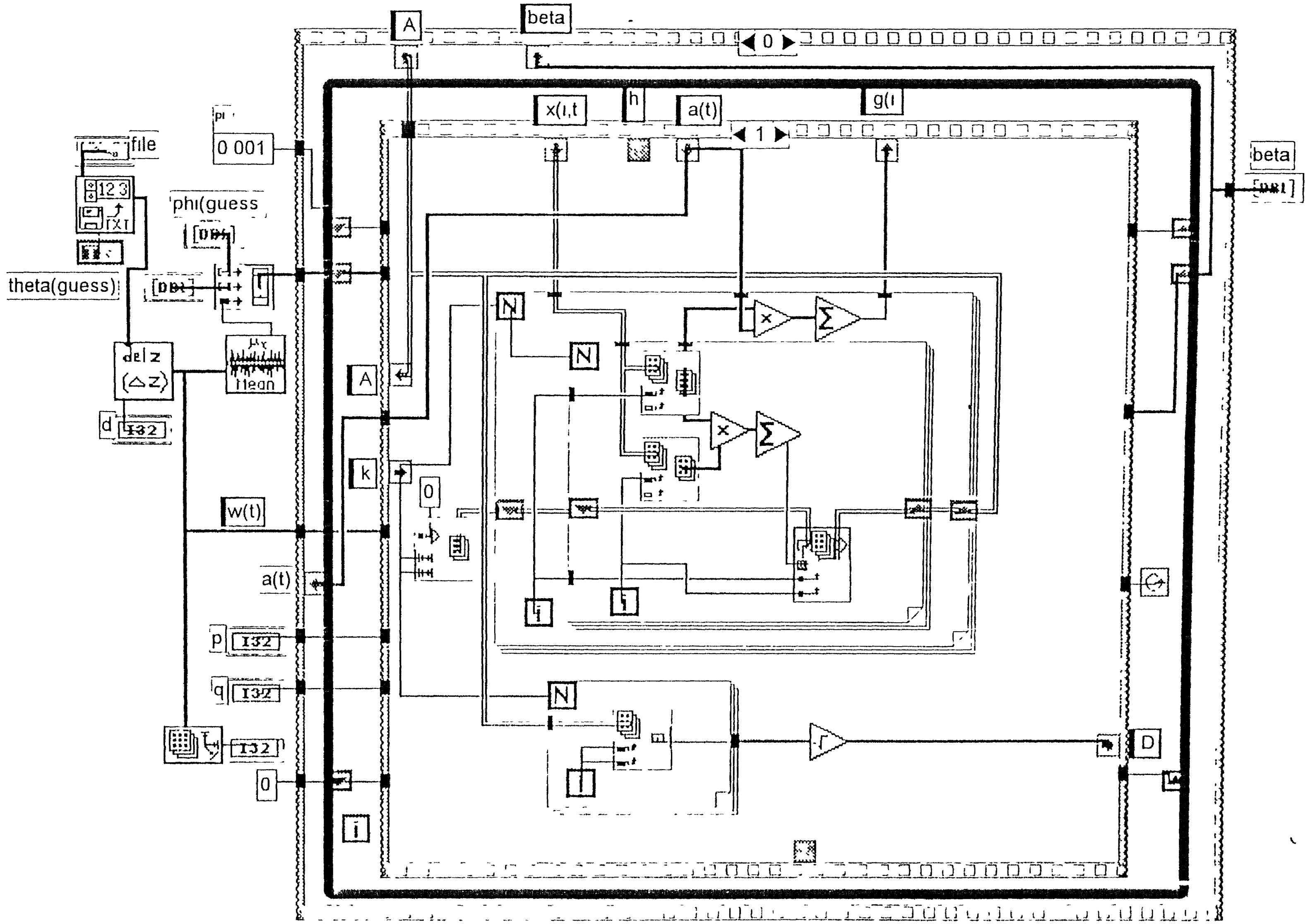


Fig. 3.22(b) VI for calculating final parameter estimation

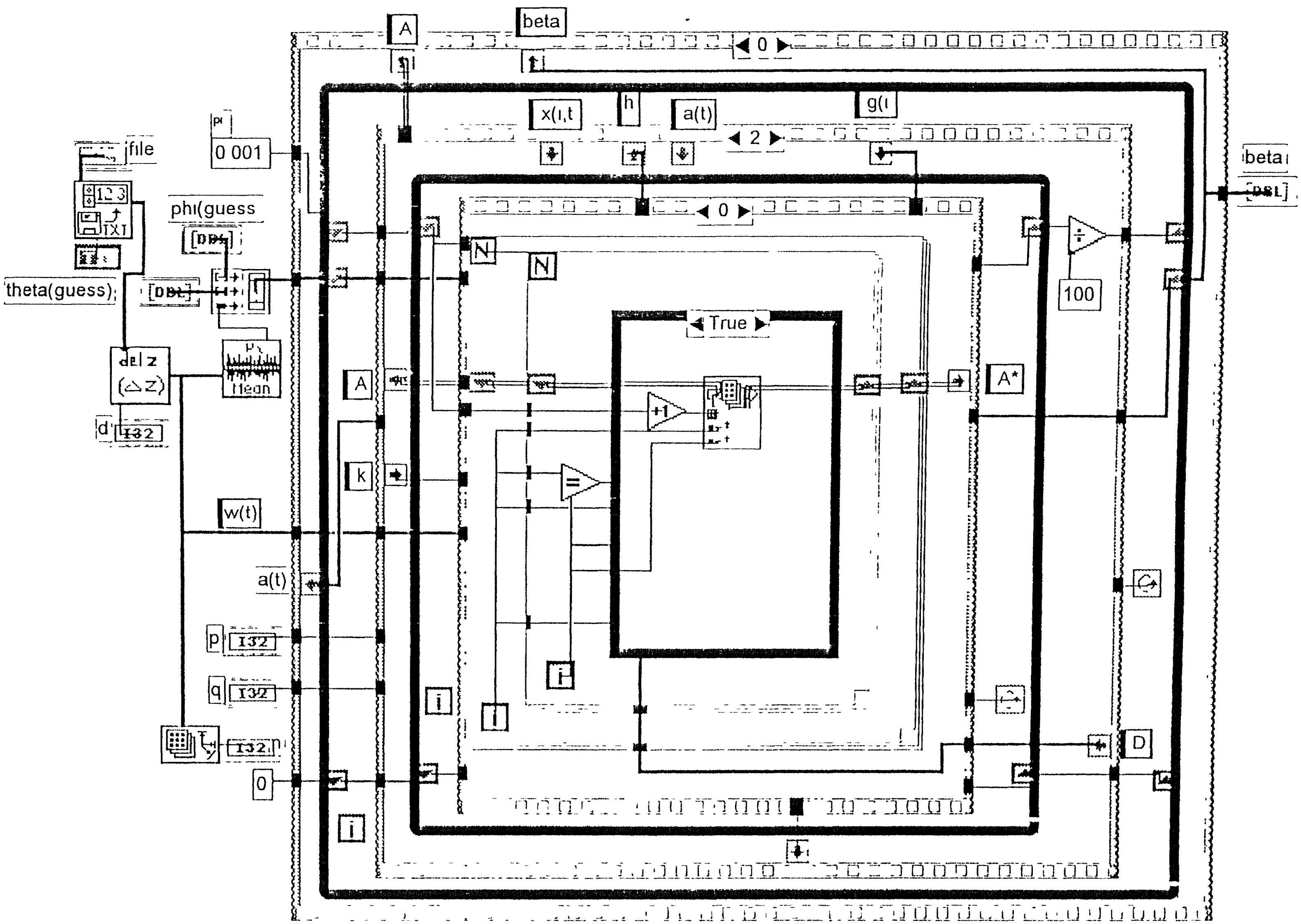


Fig. 3.22(c) VI for calculating final parameter estimation

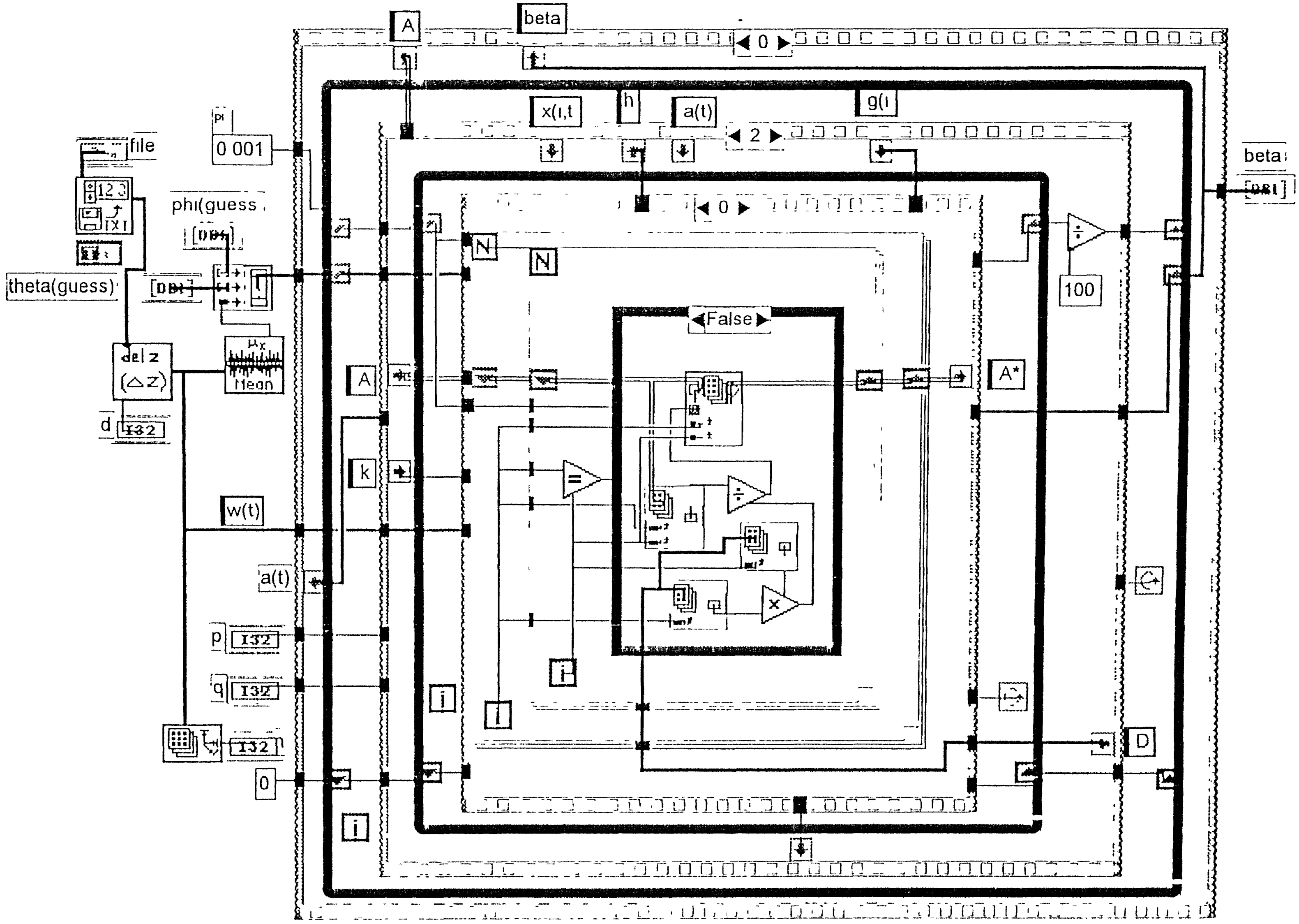


Fig. 3.22(d) VI for calculating final parameter estimation

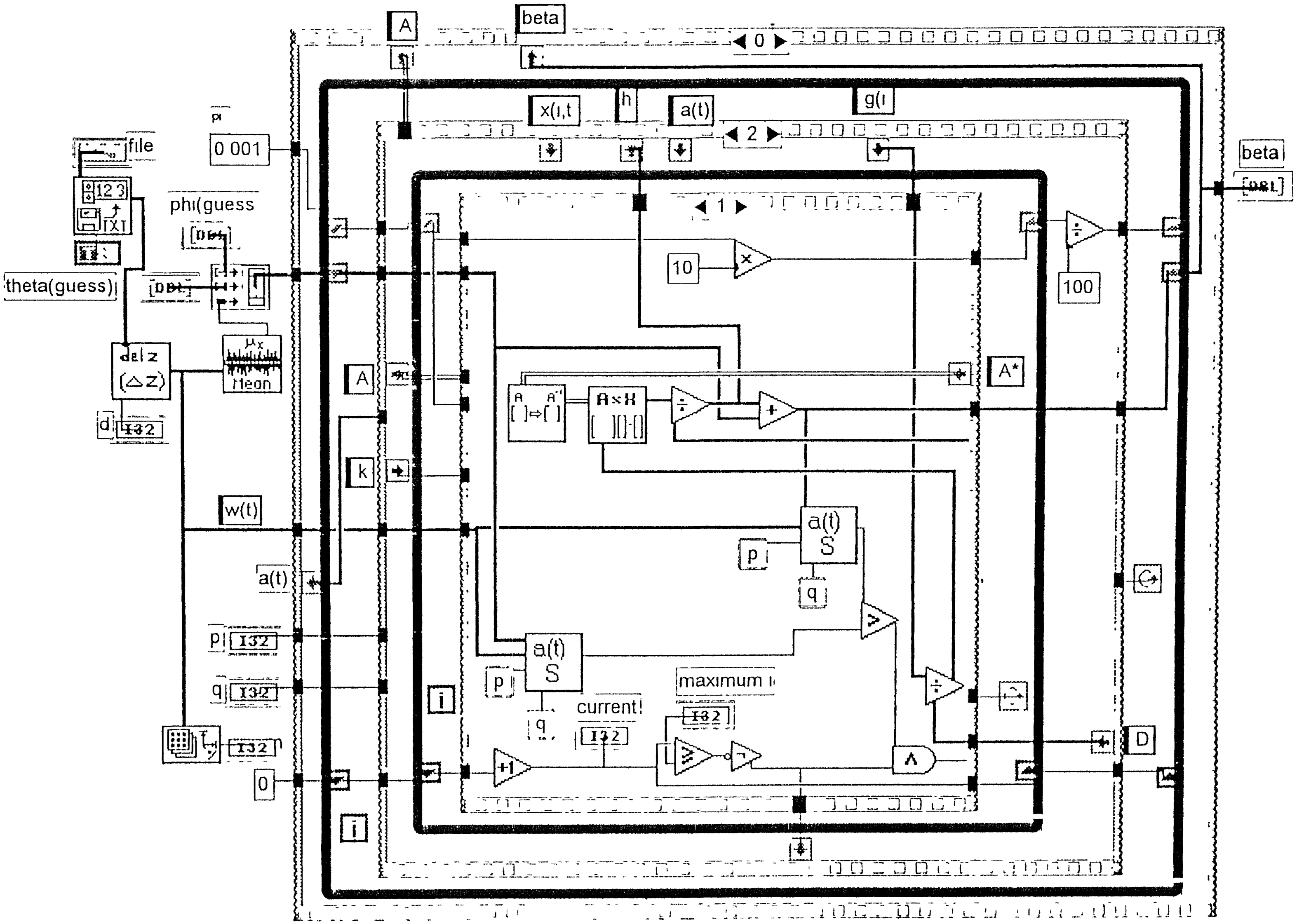


Fig. 3.22(e).VI for calculating final parameter estimation

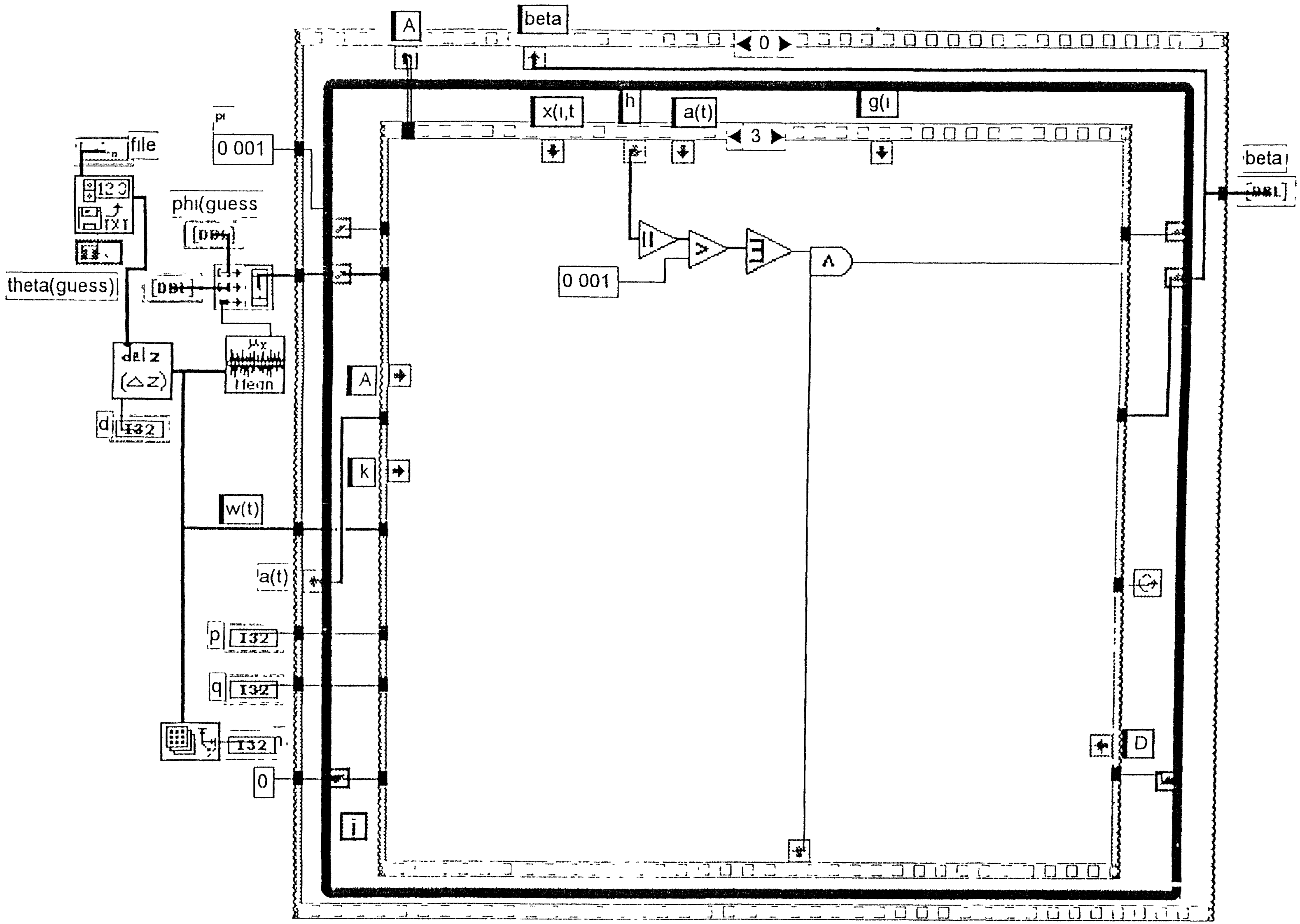


Fig. 3.22(f) VI for calculating final parameter estimation

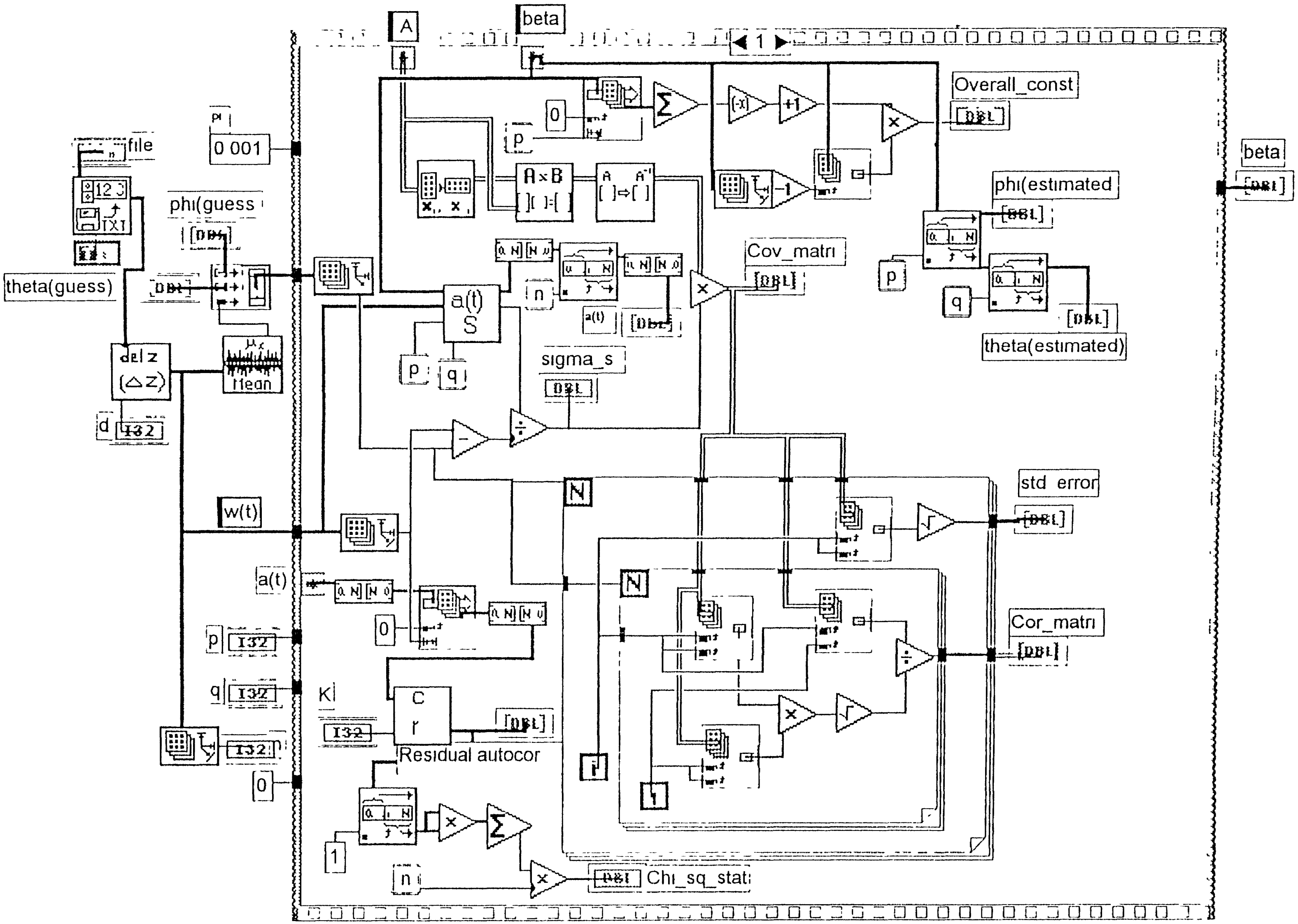


Fig. 3.22(g) VI for calculating final parameter estimation

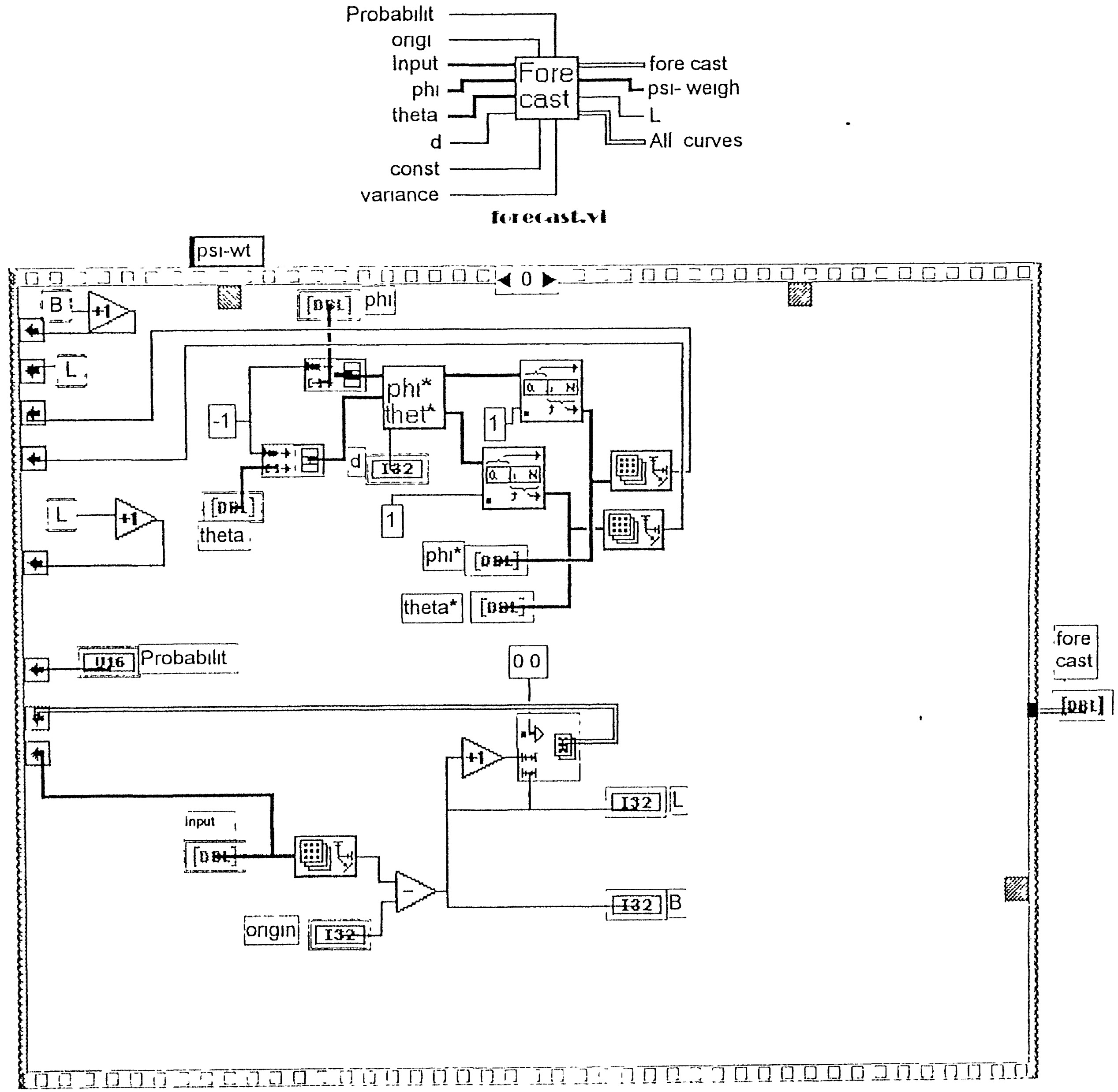


Fig. 3.23(a) VI for forecasting the final model

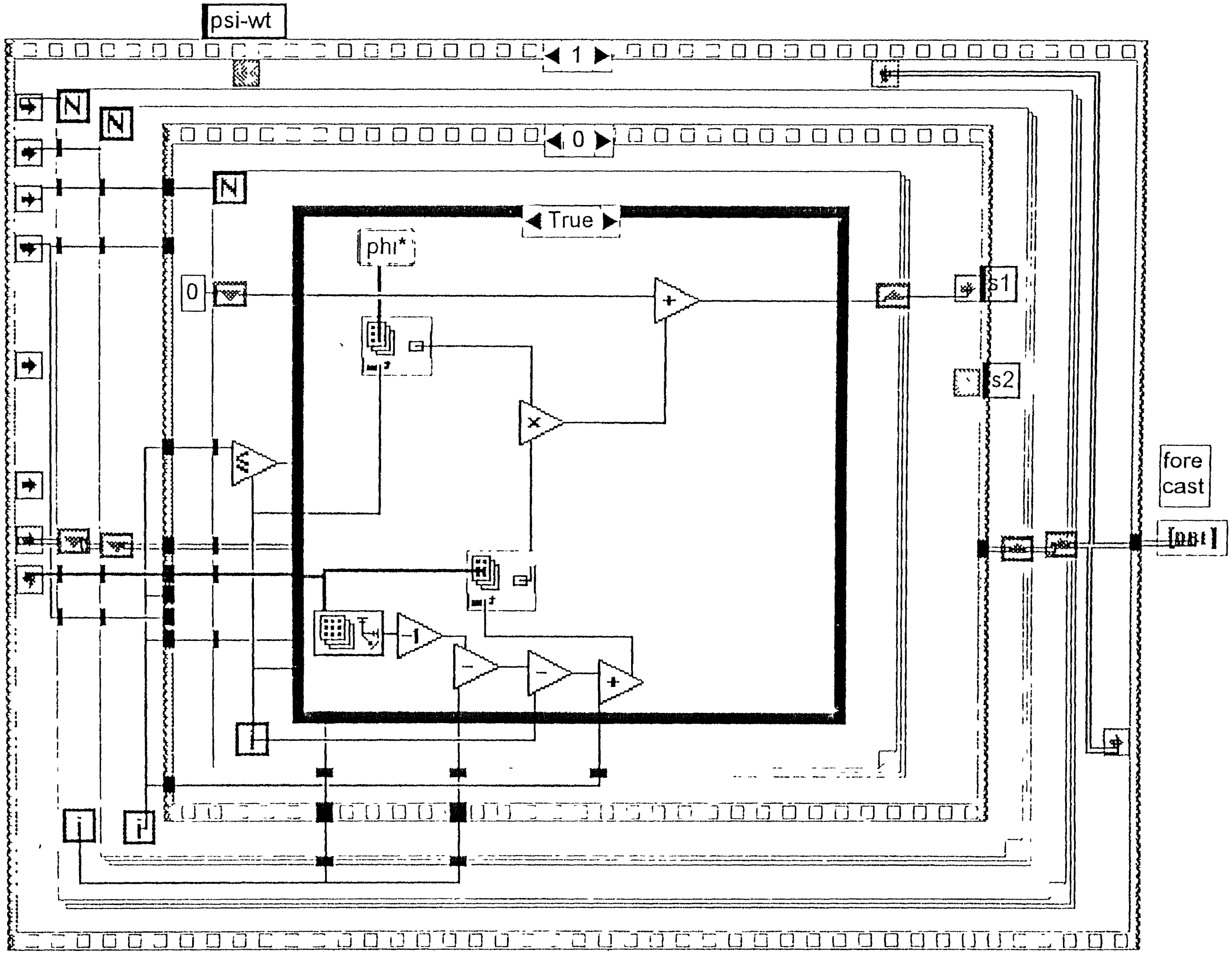


Fig. 3.23(b) VI for forecasting the final model

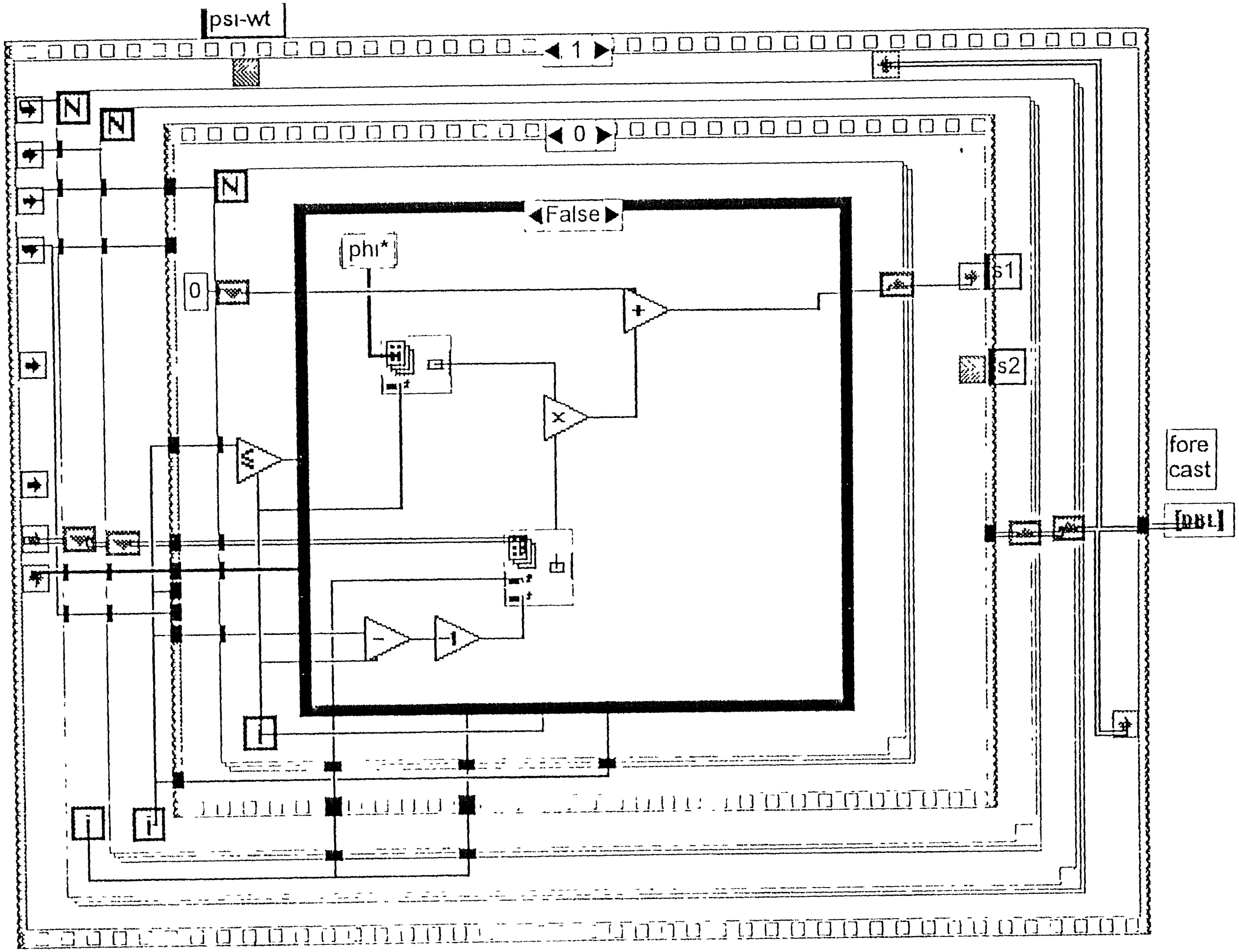


Fig. 3.23(c) VI for forecasting the final model

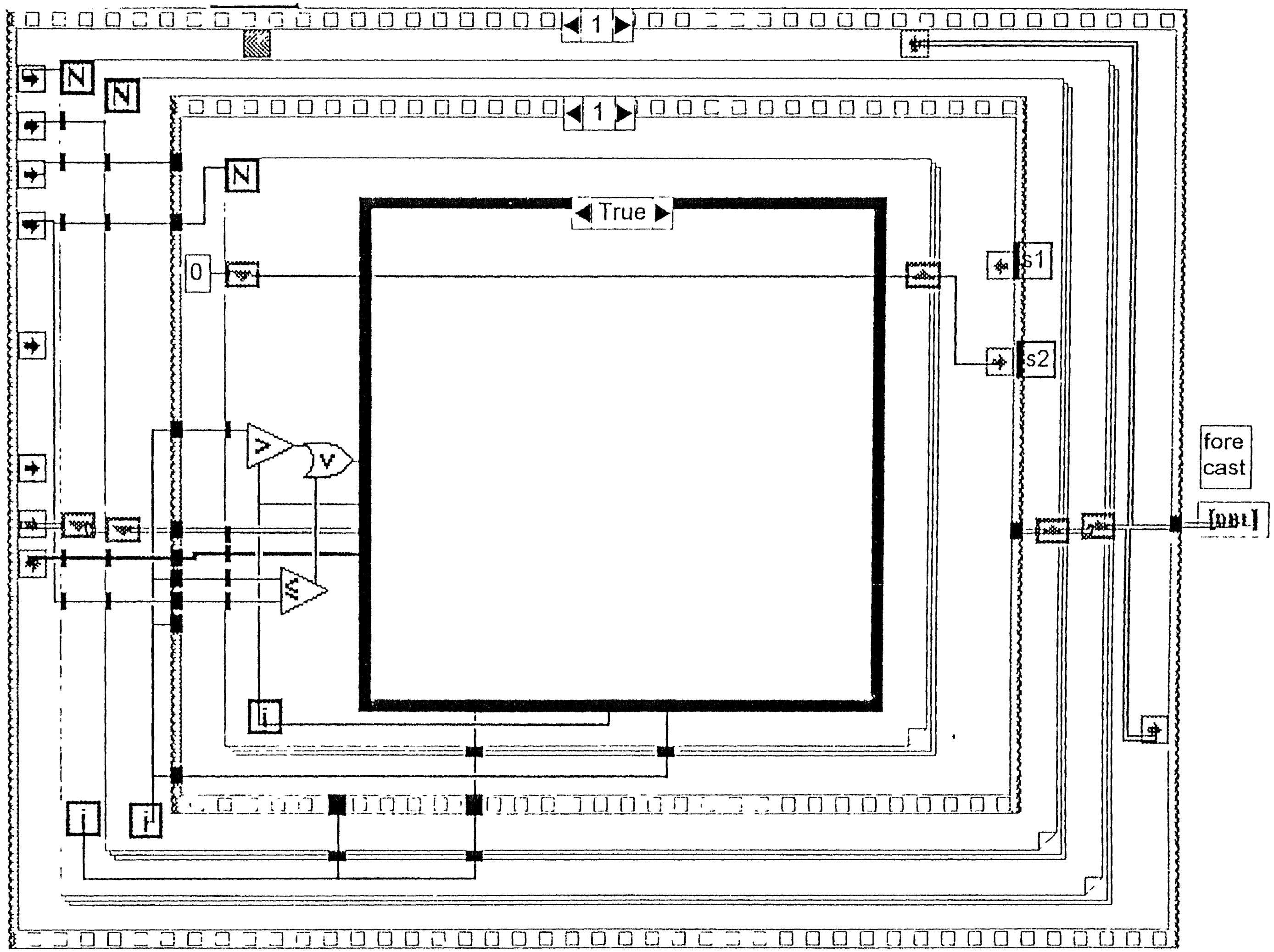


Fig. 3.23(d) VI for forecasting the final model

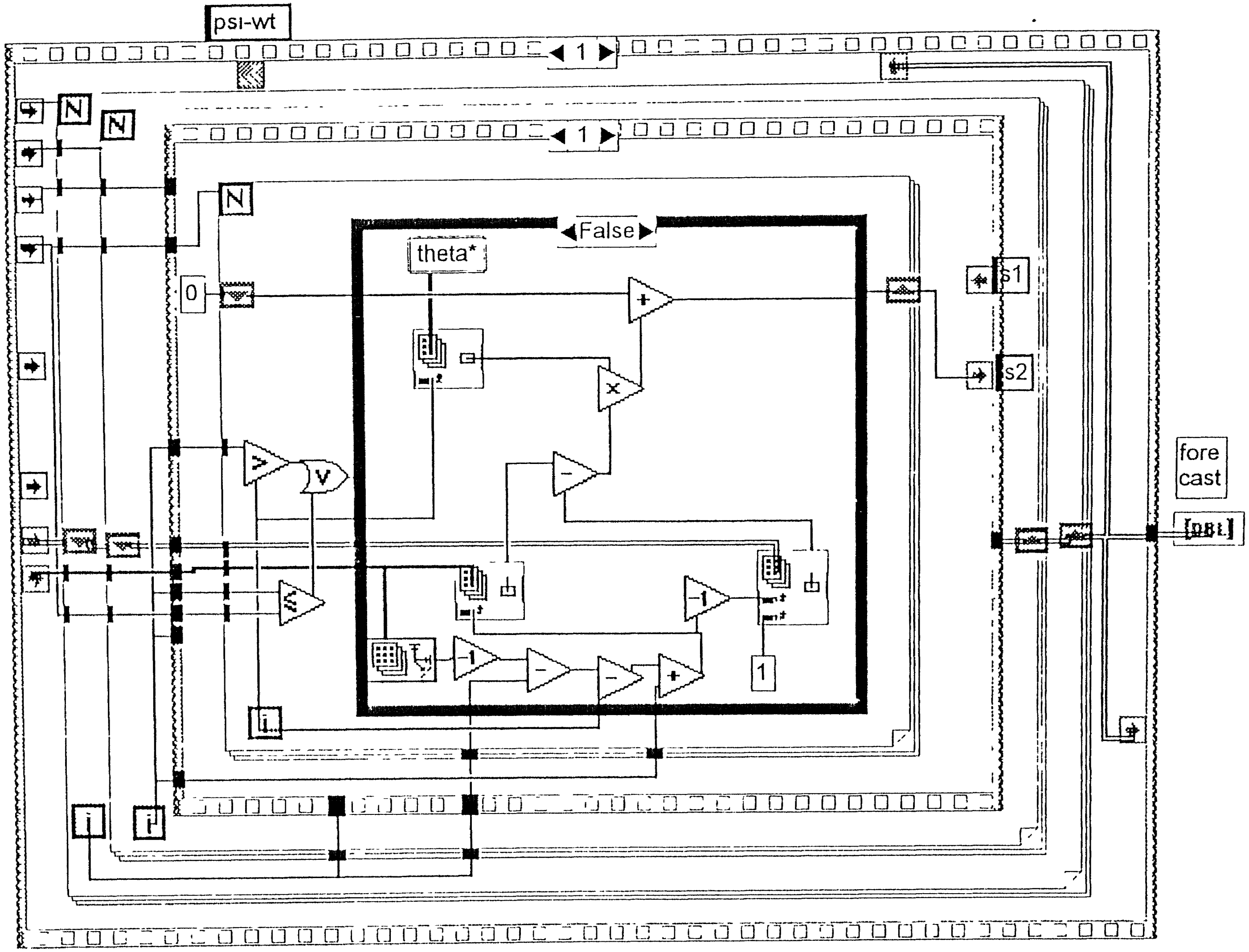


Fig. 3.23(e) VI for forecasting the final model

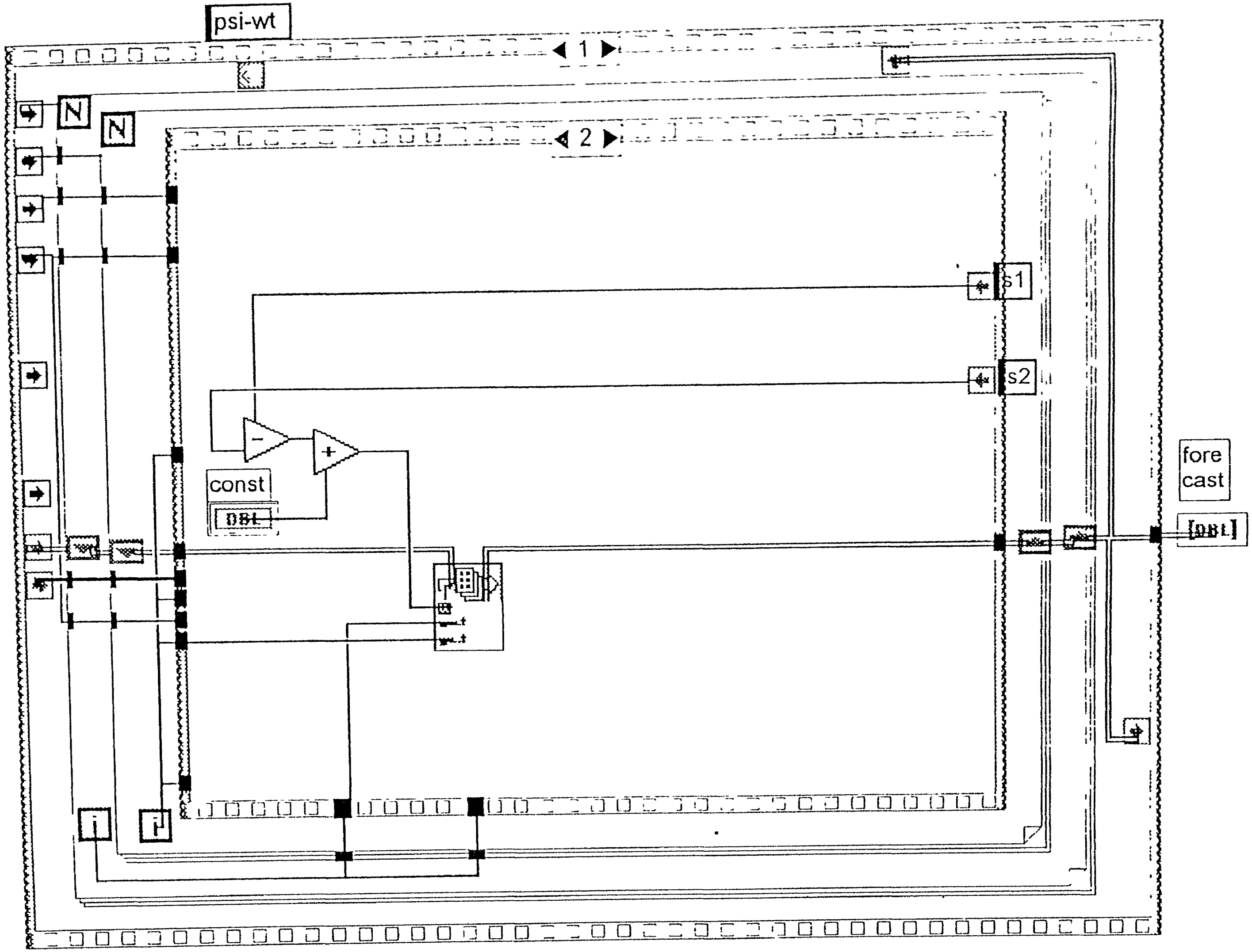


Fig. 3.23(f) VI for forecasting the final model

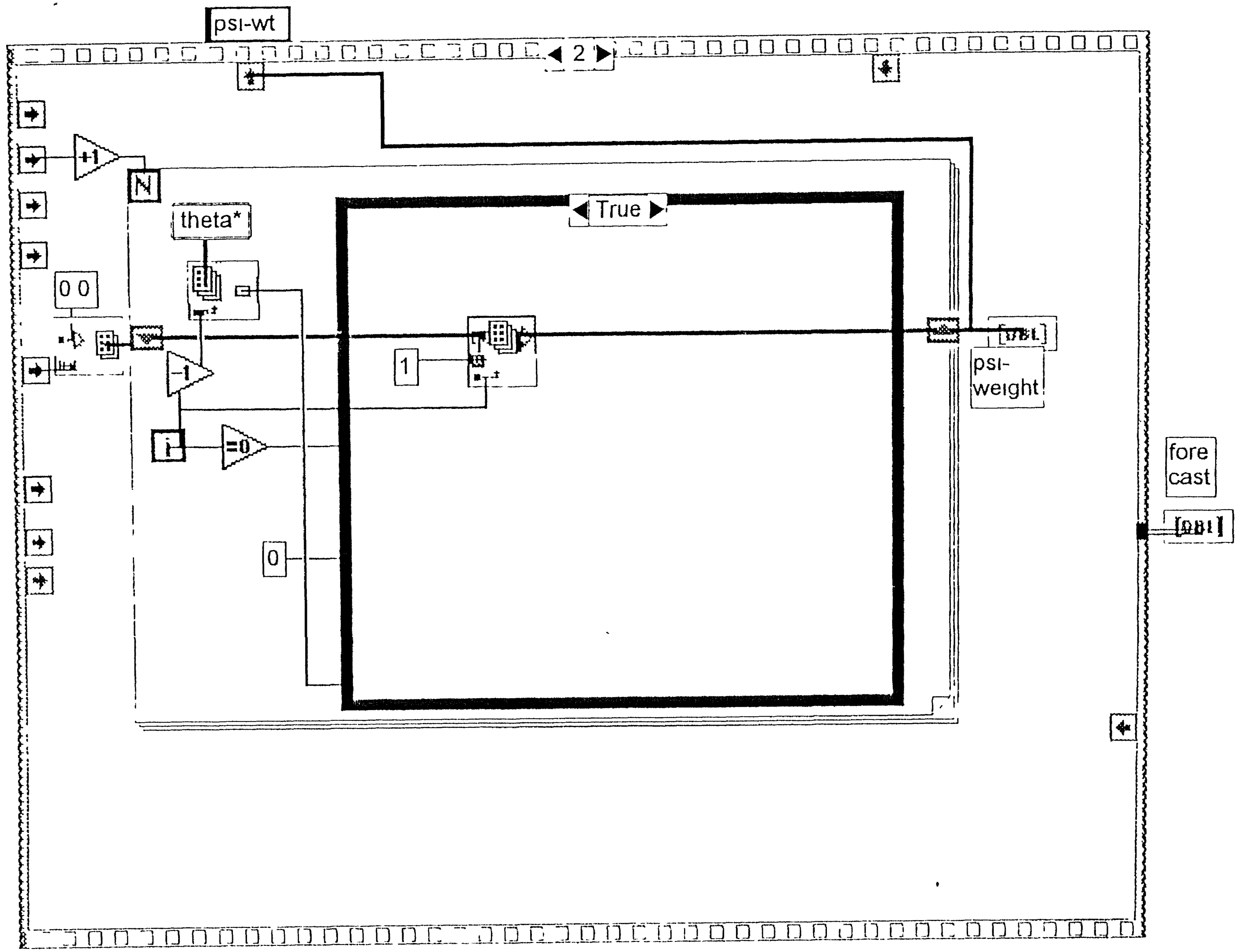


Fig. 3.23(g) VI for forecasting the final model

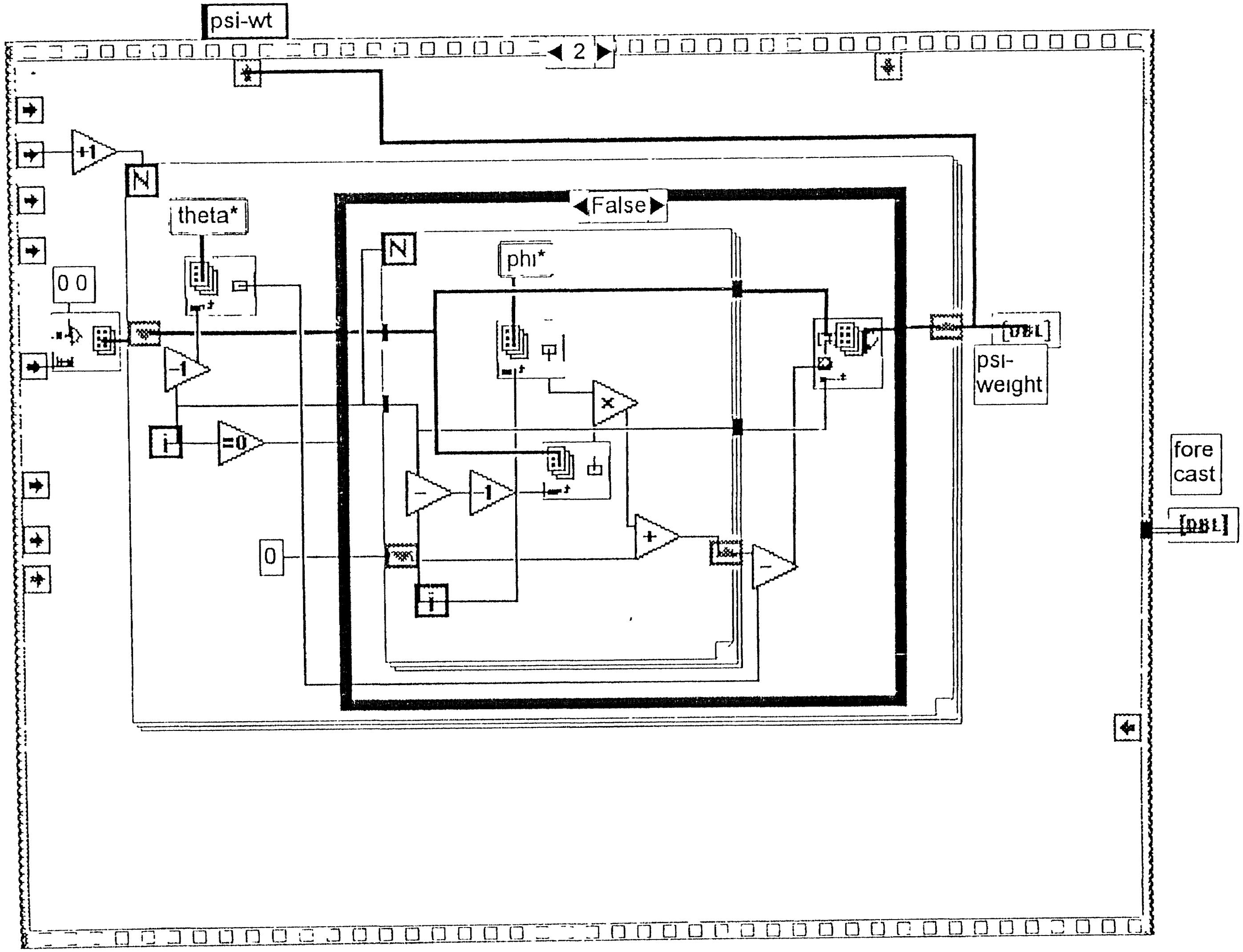


Fig. 3.23(h) VI for forecasting the final model

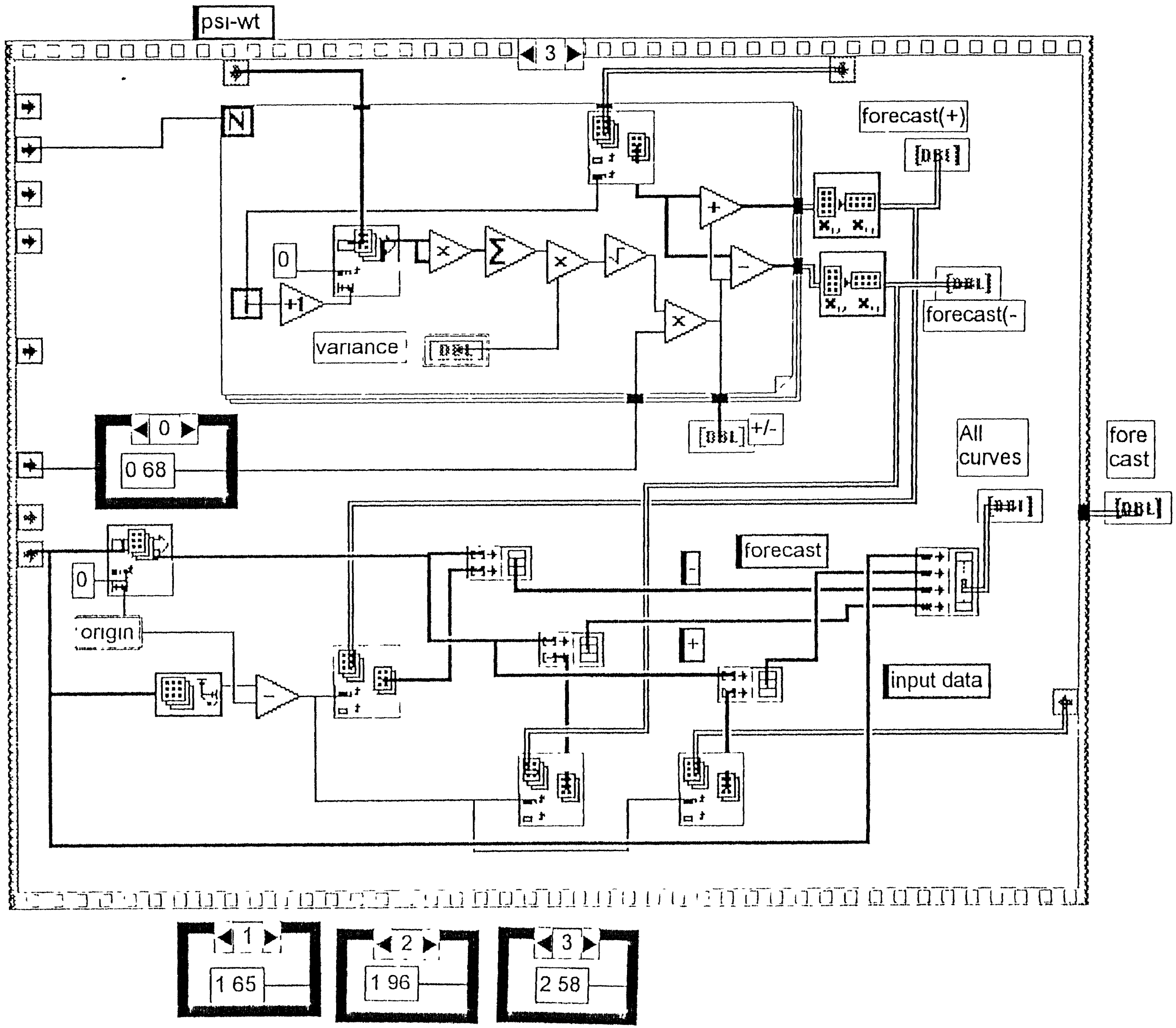


Fig. 3.23(i) VI for forecasting the final model

CHAPTER 4

CONCLUSIONS AND SCOPE FOR FUTURE WORK

During the course of the present work Virtual Instruments for Time Series Analysis have been developed. With reference to Figure 1.2, the VIs developed incorporate -

- Model Identification
- Parameter Estimation
- Forecasting

The programs have been established and tested for data available in literature.

The remaining tasks of the proposed work are as follows to make a complete Virtual Instrumentation package for on-line time series analysis .

Diagnostic Checking

The identified and estimated model is to next subjected to diagnostic checks and tests of goodness of fit with the observed data. The question of adequacy of the model is to be addressed in this stage. The inadequacies need to be defined in statistical terms and the model needs to be refined further to accommodate reasonably large body of data spread over a reasonable time span. Standard techniques of diagnostic checking involving overfitting, autocorrelation check and cumulative periodogram check are proposed to be built into the Virtual Instrumentation.

Noise Considerations

The problem of model building through a regressive analysis or transfer functions becomes complex in the presence of noise in various input/output parameters. Error estimation algorithms can be built into the models to check their robustness and to choose the model with the smallest possible mean square error.

On-line Data Acquisition From The Rotor-Rig

Certain simple instruments like oscilloscopes, FFT Analyzers and Filters were developed during the present work in order to check the on-line functioning of the AD card and its compatibility with the LabVIEW software. Vibration data was taken from the rotor-rig and displayed successfully on the above VIs. The Time Series Analysis model could not be tested on-line due to time constraints. Once the model is complete, by incorporating the Noise Considerations described above, it can be made functional to take the data from the rotor-rig.

REFERENCES

1. Barnad, G.A., "Statistical Inference," Jnl. Royal Stat. Soc., **B11**, 116, 1949.
2. Bartlett, M.S., "On the theoretical specification of sampling properties of autocorrelated time series," Jnl. Royal Stat. Soc., **B8**, 27, 1946.
3. Birnbaum, A., "On the foundations of statistical inference," Jnl. Amer. Stat. assoc., **57**, 269, 1962.
4. Box, G.E.P. and Jenkins, G M., Time Series Analysis, forecasting and control, Holden-Day, San Francisco, 1976.
5. Childs, D., 1993, Turbomachinery rotordynamics, phenomenon, modeling and analysis, John Wiley & Sons, New York.
6. Collacott, R.A., 1987, Mechanical fault diagnosis, Chapman and Hall, London.
7. Doob, J.L., Stochastic Processes, John Wiley, New York, 1953.
8. Fisher, R.A., Statistical Methods and Scientific Inference, Oliver and Boyd, Edinburgh, 1956.
9. Grenander, U. and Rosenblatt, M., Statistical Analysis of Stationary Time Series, John Wiley, New York, 1957.
10. Hannan, E.J., Time Series Analysis, Mathuen, London, 1960.
11. Kendall, M.G., " On the analysis of oscillatory time series," Jnl. Royal Stat. Soc., **108**, 93, 1945.
12. John Kolmogoroff, A, "Stationary sequences in Hilbert space," Bull. Math. Univ., Moscow 2, No. 6, 1941.
13. Marquardt, D.W., "An algorithm for least squares estimation of nonlinear parameters, "Jnl. Soc. Ind. Appl. Math., **11**, 431, 1963.
14. Mitchell, J., 1981, Machinery analysis and monitoring, PennWell Publishing Company, Oklahoma.
15. Quenouille, M..H., Analysis of Multiple Time Series, Hafner, New York, 1957.
16. Quenouille, M..H., Associated Measurements, Butterworth, London, 1952.
17. Rao, J.S., 1992, Rotor Dynamics, Wiley Eastern Ltd., New Delhi.
18. Robinson, E.A., Multichannel Time Series Analysis, Holden-Day, San Francisco, 1967.
19. Walker, G, "On periodicity in series of related terms," Proc. Royal Soc., **A131**, 18, 1931.
20. Whittle, P., Prediction and Regulation by Linear Least-Squares Methods, English Universities Press, London, 1963.
21. Wiener, N., Extrapolation, Interpolation and smoothing of Stationary Time Series, Wiley, New York, 1949.
22. Wilson, G.T., "Factorization of generating function of a pure moving average process," SIAM Jnl. Num. Analysis, **6**, 1, 1969.
23. Wold, H.O., A Study in The Analysis of Stationary Time Series, Almquist and Wicksell, Uppsala, 1938 (2nd. ed. 1954).
24. Yule,G.U., " On a method of investigating periodicities in disturbed series with special reference to Wolfer's sunspot number," Phil. Transe., **A226**,267, 1927.